28234–6027–RU–00

(NASA-CR-147755) LACIE PERFORMANCE PREDICTOR FINAL OPERATIONAL CAPABILITY PROGRAM DESCRIPTION, VOLUME 1 (TRW SYSTEMS GROUP) 339 p HC $10.00 CSCL 02C

N76-25630

UNCLAS

G3/43 42116

NASA CR-147755

LACIE PERFORMANCE PREDICTOR FINAL OPERATIONAL CAPABILITY PROGRAM DESCRIPTION

VOLUME I

MAY 1976

LARGE AREA CROP INVENTORY EXPERIMENT

LACIE

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lyndon B. Johnson Space Center
Houston, Texas

Contract Number NAS-9-14547

TRW
SYSTEMS GROUP
ONE SPACE PARK · REDONDO BEACH · CALIFORNIA 90278

This document, in three volumes, describes the FOC version of the LACIE Performance Predictor produced under Contract NAS9-14547. NASA/JSC Contract Technical Monitor is I. D. Browne, Earth Observations Division.

Table of Contents

EPHEMS BOOK I

Table of Contents

PART I

PROBLEM DESCRIPTION

PROBLEM DESCRIPTION FOR EPHEM PROGRAM

1.0 SCOPE

1.1 PROGRAM CAPABILITIES

This program will compute the orbital parameters for up to 2 vehicles orbiting about the earth for up to 549 days. This data is sorted on disk files and represents a continuous swath path about the earth. These swath tables are to be used by the segment access generator (SAGE) program to determine when and if certain target land segments lie in the path of the swath.

1.2 METHOD OF PROGRAM DEVELOPMENT

Use will be made of existing FORTRAN subroutines in order to compute the ephemeris data and the data for the swath tables. Except for a limited number of specific places, the equation processing portion of these routines will remain intact. Control logic will be altered but new routines will only have to be created for processing card input and writing the swath files.

1.3 OPERATIONAL ASSUMPTIONS

- There can be a maximum of 2 vehicles.
- There can be a maximum of 549 days.
- The latitude band for swath table generation is $\pm 65^{\circ}$ latitude and there will be a maximum of 100 latitude points to process.
- Only 1 case is run at a time.
- The complete card data set must be entered for each run.
- The existing swath tables will be destroyed each time this program is run.
- The swath tables will be designated as permanent files since it is assumed that this program will be run about once per year.
- Since namelist is not available on the PDP, fixed field card formats are utilized.

- A sun synchronous inclination will be computed for each run condition
- Only first order oblateness terms are modeled in the orbit generation, i. e., nodal regression and apsidal precession.
- Orbital elements are maintained as Keplerian elements (a, e, i, Ω, w, t) and are updated on a rev by rev basis.
- Orbit initialization equations will insure passage through a reference point at a specified time on the first day. Active passage is assumed to be on the descending pass.

2.0 INPUT

The only input media to this program are cards. The following sections give the list and definitions of the input quantities, the card formats and rules of entry.

2.1 LIST OF INPUT QUANTITIES

See input data description sheets on pages 6 and 7.

2.2 CARD FORMATS

See card format sheet on page 8.

2.3 DECK SETUP

The order of input is:

1. Header card
2. Program control card
3. Orbital determination data (2 cards)

2.4 RULES FOR ENTERING DATA ON CARDS

2.4.1 General

1. Integers must be entered right justified.
2. F-format numbers must have the decimal point present, i.e., F5.1-XXX.X, F4.0-XXX.
3. The card sequence numbers in the C.C. 79-80 must be present in all data cards.

2.4.2 Rules for Specific Fields

IVEH - No more than 2 vehicles can be entered in this field.

NODAY - A number larger than 549 will not be accepted.

IYR - An entry less than 64 will not be accepted.

INLAT and ISLAT - A negative entry or an entry greater than 65 will not be accepted. The total number of latitudes allowed, represented by the combined bands, is 100.

IPPI5 - An entry larger than 549 will not be accepted.

SA - An absolute value greater than 10° is not allowed.

IV1TIM and IV2TIM - The first subscript entry (year) must be between 1964 and the present.

Input Data Description

Program Control Input

Name	Symbol	Dimension	Nominal Value	Range	Units	Description
ICASE		1	0	1-9999	--	A 4 digit case number to identify the printed output and the swath tables
NODAY		1	549	1-549	--	No. of days to generate ephemeris data for the swath table
IVEH		1	2	1-2	--	Number of vehicles to process
HEADER		8	Blanks	---	--	Provision for an 72 character case header to print out at the top of every output print page
LIDEBG		4	F (Blank)	T or F	--	Flag to allow printout of intermediate data for swath calculations. Data for the following rouitnes is printed based on subscript values: 1. SWATH 2. Swath table and reference file 4. REVTAB
INLAT		2	5, 65	0-65	Deg	Northern Hemisphere latitude band for swath generation
ISLAT		2	15, 45	0-65	Deg	Southern Hemisphere latitude band for swath generation
IPPI5(10)		10	1, 549 8*0	1-549	--	Ephemeris data display flag = [start day, end day] pairs
ISOSTR		1	6	0-12	--	No. of months delay to start the generation of swath data on the swath table for the southern latitudes

Orbital Determination Input

Name	Symbol	Dimension	Nominal Value	Range	Units	Description
IV1TIM		6	0	→	→	Vehicle 1 reference latitude passage time IV1TIM(1)-(Year-1900); IV1TIM(2)-Month r IV1TIM(3)-Day No.; IV1TIM(4)-Hours IV1TIM(5)-Minutes; IV1TIM(6)-Seconds
IV2TIM		6	0	→	→	Vehicle 2 reference latitude passage time; same information as Vehicle 1.
ORBIT1		6	0			Orbital elements for Vehicle 1
(1)	a			6650 to 7700	Kilom.	Semi-major axis
(2)	e			10^{-5}<.15	--	Eccentricity
(4)	Ω			0-360	Deg.	Longitude of Reference Latitude
(5)	ω			0-360	Deg.	Argument of perigee
(6)	λ			0- $\pm 65^{o}$	Deg.	Vehicle reference latitude
ORBIT2		6	0			Orbital elements for Vehicle 2. Same data as for Vehicle 1.
SA		3	0	-10-+10	Deg.	Reference scan angles defining swath field of view

PROGRAM CONTROL CARD FORMAT

C.C.	1	5	7	10		18	11	22	25	28	31	34	37	40	43	46	49	52	79
				Lat. Band				Ephemeris Data Display Flag Pairs											
								IPPI5(1), (2)		IPPI5(3), (4)		IPPI5(5), (6)		IPPI5(7), (8)		IPPI5(9), (10)			
	ICASE	IVEH	NODAY	INLAT	ISLAT	L1DEBG		Start Day	Stop Day	Start Day	Stop Day	Start Day	Stop Day	Start Day	Stop Day	Start Day	Stop Day	ISOSTR	01
	I4	I2	I3	2I2	2I2	L4		I3	I3	I3	I3	I3	I3	I3	I3	I3	I3	I2	I2

ORBITAL DETERMINATION DATA CARD FORMATS

CARD 1

C.C.	1	6	11	16	18	20	22	24	26	28	30	32	34	36	38	79
	SA(1)	SA(2)	SA(3)	IV1TIM - Veh. 1						IV2TIM - Veh. 2						
	Left Side Refer. Scan Angle	Veh. Refer. Scan Angle	Right Side Refer. Scan Angle	Year	Month	Day	Hour	Minute	Seconds	Year	Month	Day	Hour	Minute	Seconds	02
	F5.1	F5.1	F5.1	I4	I2	I3	I2	I2	I2	I4	I2	I3	I2	I2	I2	I2

Data Types

- I - Integer
- AN - Alphanumeric
- F - Floating Decimal
- L - Logical (T or F)

CARD 2

C.C.	1	9	15	19	25	29	35	43	49	53	59	63	79
	ORBIT1 - Veh. 1						ORBIT2 - Veh. 2						
	a	e	X	Ω	ω	τ	a	e	X	Ω	ω	τ	03
	F8.3	F6.5		F6.2	F4.0	F6.2	F8.3	F6.5		F6.2	F4.0	F6.2	I2

3.0 PROCESSING

3.1 OVERVIEW

An existing analytic ephemeris program is being used to simulate the orbital process which maintains the data as follows:

(1) A set of Keplerian orbital elements for each rev (a, e, i, Ω, ω, t).

(2) Ground path and swath visibility tables as a function of latitude to be used by SAGE in determining access conditions.

The only original code for this program will be related to control logic and orbit initialization procedures.

3.2 PROGRAM FLOW

The subroutine MAIN will be used as the master control for this program. Its logic flow and functions are shown in the MAIN block diagram on Page 11. This diagram illustrates the usage of the major subroutines. The methodology for the generation of ephemeris data is performed in HECTOR and its flow is shown on pages and . Subroutine ACCGEN controls the computation of swath data for one revolution. The list of major routines performing mathematical calculations are:

1. HECTOR (Called by MAIN)
   Computes for each revolution:
   - Semi-major axis
   - Eccentricity
   - Inclination
   - Longitude of the node
   - Argument of perigee
   - Time of perigee passage
   - Start time of rev
   - Right ascension of the node

2. LFPA - entry ALPHA (Called by HECTOR).
   Computes right ascension of true and mean sun.

3. KEPLER (Called by HECTOR)
   Computes the eccentric anomaly by iteration technique, i.e., solves for E in $\{ E - e \sin E \equiv n \; \Delta t \}$

4. ACCGEN
   Computes eccentric anomaly at ascending node - E. Call SWATH and REVTAB to generate swath data for one revolution.

5. SWATH
   Generates the latitude and delta longitude data for the scan angles representing a swath path about the earth as a function of eccentric anomaly.

6. REVTAB
   Reformats the data from the SWATH routine data to represent this data at discrete 1^{o} latitude increments.

3.2.1 Initialization Procedures to Ensure a Sun Synchronous and Reference Passage Condition (LACIE Requirement)

(a) The input inclination is overriden in INIT by a computed inclination which ensures a sun synchronous orbit. A sun synchronous orbit is defined as an orbit whose nodal regression rate matches the mean sun motion per day:

Recompute i $\equiv$ (ORBIT1(3) and ORBIT2(3) to make the orbit sun synchronous.

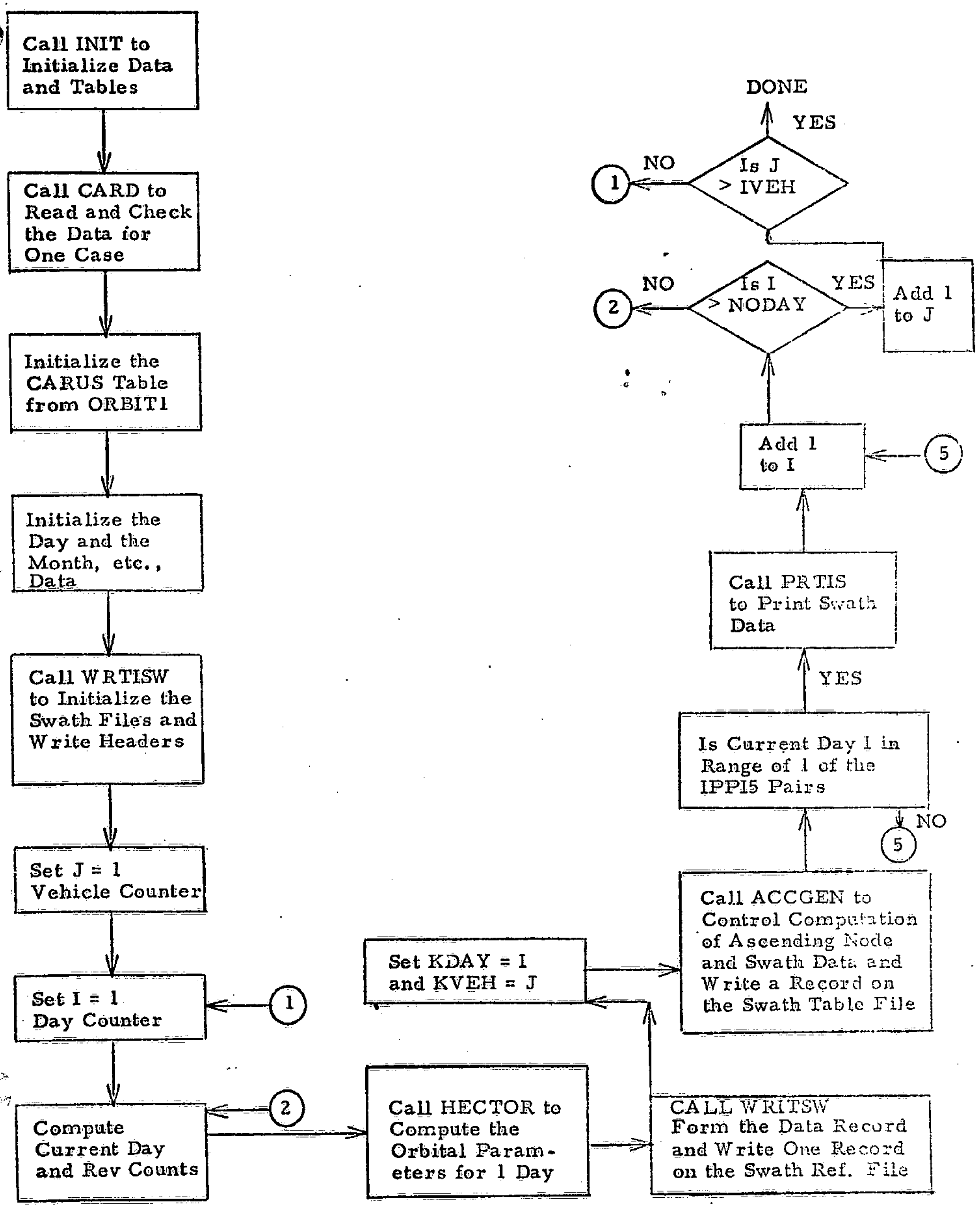
$$i = \cos^{-1} \left[\frac{-4\pi \, (a^{2}) \, (1 - e^{2})^{2}}{3 * (CMU/_{a}{}^{3})^{1/2} * CJ2X365.25*86400} \right] \quad (1)$$

(b) An initial rev node, Ω node, and time of node, t node, will be computed from the input (λ, $\emptyset$) [reference point] and passage time IV1TIM, IV2TIM) for each active vehicle. The procedure is:

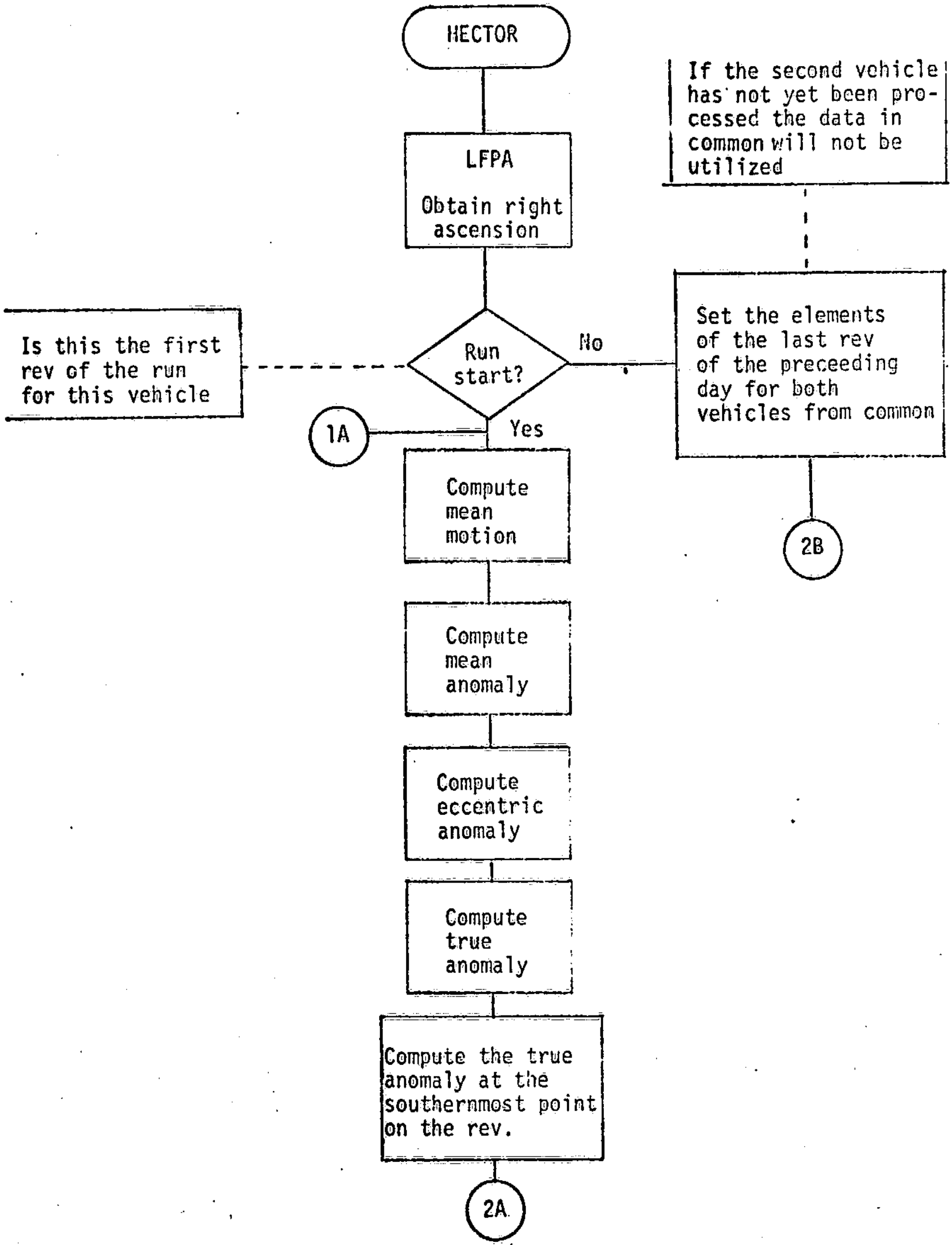
(1) Back off K1, K2 revs such that IV1TIM, IV2TIM remain within current day.

Program Flow for Ephemeris Program

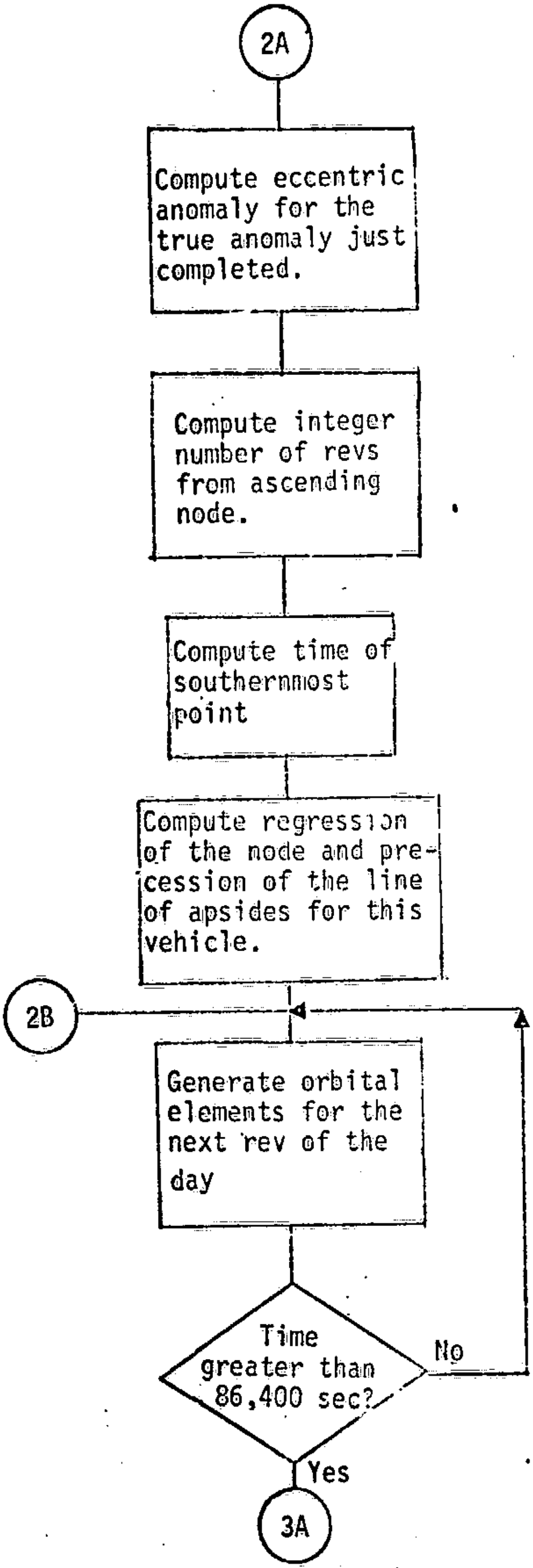
Master Control - MAIN



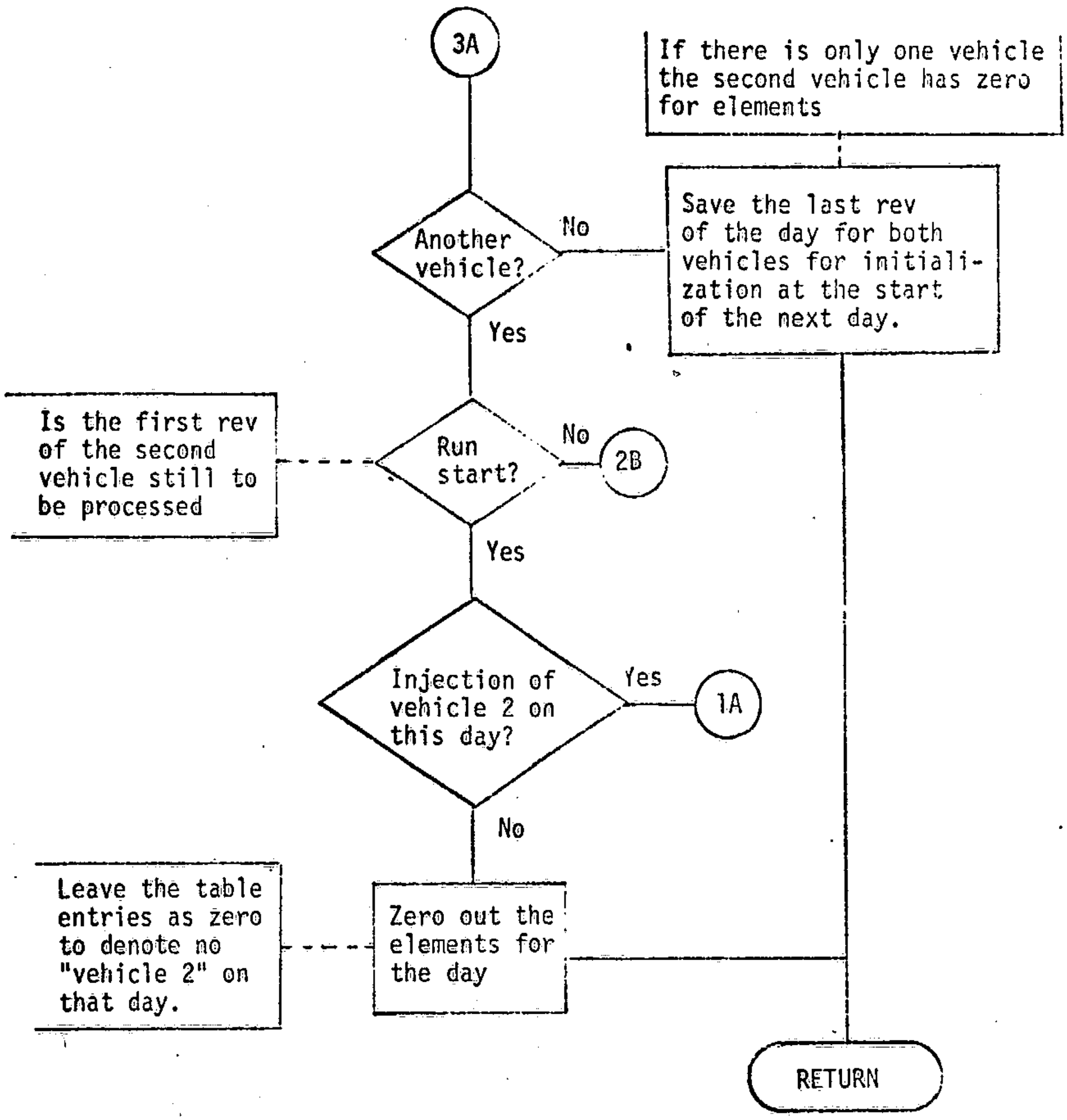
EPHEMERIS FLOW



EPHEMERIS FLOW (CONT'D)



EPHEMERIS FLOW (CONT'D)



(2) Compute initial rev node by

$$\Omega_{node} = \emptyset_{input} + \pi - \mathrm{Sin}^{-1}\left(\frac{-\tan\lambda}{\tan i}\right) + K_i\, \Delta\Omega_{rev} \qquad (2)$$

where,

$$\Delta\,\Omega_{rev} = \frac{4\pi^2}{86400\,\overline{n}}$$

$$\overline{n} = n_o\left[1 + \frac{3}{2}\, J_2\, \sqrt{\frac{1 - e^2}{p^2}}\, \left(1 - \frac{3}{2}\, \sin^2 i\right)\right]$$

$$n_o = \sqrt{Mo/a^3}$$

$= $ mean motion

$$p = a(1 - e^2)$$

$\mu_o =$ gravitational constant

$a =$ semi-major axis

These items were incorporated into the HECTOR initialization procedure. The mean motion calculation, $\overline{n}$, was implemented by perturbing μ calculation.

3.2.2 Computation of Orbital Elements for Initial Rev of Simulation

HECTOR will obtain the right ascension of Greenwich for the initial rev of the day. At the start of the run HECTOR will utilize the injection conditions for each vehicle to compute the elements at the start of the rev. The following equations will be utilized for computation of the elements of the first rev of the simulation.

$$\mu = \mu_o K^2$$

where,

$$K = 1 + \frac{3}{2} J_2 \sqrt{\frac{1 - e^2}{p^2}}\, \left(1 - \frac{3}{2}\, \sin^2 i\right) \qquad (3)$$

$$\text{Mean Motion} = n = \sqrt{\mu/a^3} \tag{4}$$

$$\text{Mean Anomaly} = n\ (t_o - T) = M \tag{5}$$

where

$a \equiv$ semi-major axis

$t_o \equiv$ time of injection in seconds from midnight

$T \equiv$ time of perigee passage

and

$$t_o > T \text{ (if } T > t_o \text{ then } T = T - \frac{2\pi}{n}\text{)}$$

The eccentric anomaly, E, is defined by

$$E - e \sin E = M \tag{6}$$

and the solution for E is obtained from the following iterative technique with ϵ defined as the required accuracy for E.

(a) assume E_o and compute M_o

(b) compute $\Delta M = M - M_o = n(t_o - T) - M_o$

(c) compute $1 - e \cos E_o$

(d) $\Delta E = \Delta M / (1 - e \cos E_o)$

(e) $E_1 = E_o + \Delta E$

(f) $M_1 = E_1 - e \sin E_1$

(g) $\delta M = M - M_1$

(h) if $\delta M > \epsilon$ go to (c) with $E_o = E_1$ and $\Delta M = \delta M$

The true anomaly at injection is

$$\nu = \sin^{-1}\left(\frac{\sqrt{1-e^2}\ \sin E}{1-e\cos E}\right) \tag{7}$$

and the quadrant of ν is determined from Equation (7) and that $\cos E - e < 0$ if ν is in the 2nd or 3rd quadrant since

$$\cos \nu = \frac{\cos E - e}{1 - e \cos E} .$$

The inplane angle from the ascending node to the injection point is given by

$$u = (\omega + \nu)_{\text{MOD } 2\pi} \tag{8}$$

where ω is the argument of perigee from the input.

The inplane angle from the injection point to the southernmost point on the rev (i.e. the start of the next rev) is

$$\Delta\nu = \frac{3}{2}\pi - u \tag{9}$$

and the true anomaly at the southernmost point is

$$\nu_1 = (\nu + \Delta\nu)_{\text{MOD } 2\pi} \tag{10}$$

The cosine of the eccentric anomaly at ν, is

$$\cos E_1 = \frac{1}{e}\left(1 - \frac{1-e^2}{1+e\cos\nu_1}\right) = \frac{e+\cos V_1}{1+e\cos V_1} \tag{11}$$

The eccentric anomaly at ν_1 is

$$E_1 = \sin^{-1}\left[\frac{\sin\nu_1(1-e\cos E_1)}{\sqrt{1-e^2}}\right] \tag{12}$$

with the quadrant of E_1 uniquely determined by (11) and (12).

The integer number of perigee to perigee revs prior to the southern-most point is given by

$$k = \left[(\nu + \Delta\nu)/2\pi\right]_{int} \tag{13}$$

The time at the sounthernmost point is

$$T_s = \frac{E_1 - e\sin E_1}{n} + T + k\left(\frac{2\pi}{n}\right) \tag{14}$$

where $n \equiv$ mean motion

$T \equiv$ time of perigee passage

Likewise, the time of the ascending node for thé rev is

$$T_1 = \frac{E^* - e\sin E^*}{n} + T + k\left(\frac{2\pi}{n}\right) \tag{15}$$

where $E^* = E_1 + \pi/2$

The regression of the node and precession of the line of apsides are computed from

$$\dot{\Omega} = \frac{-3}{2} \frac{n J_2}{a^2(1-e^2)^2} \cos i \tag{16}$$

$$\dot{\omega} = \frac{-3}{2} \frac{n J_2}{a^2(1-e^2)^2} \left(\frac{5 \sin^2 i}{2} - 2 \right) \tag{17}$$

where
- a ≡ semi major axis
- e ≡ eccentricity
- n ≡ mean motion
- J_2 ≡ second harmonic of potential function

If the inclination of the orbit is greater than $\pi/2$ the node does not undergo regression but advances as shown by Equation (16).

From equations (15), (16), and (17) the longitude of the node and the argument of perigee at the time of the node of the first rev can be obtained.

$$\Omega_1 = \Omega_o + (\dot{\Omega} - \omega_e)(T_1 - t_o) \tag{18}$$

$$\omega_1 = \omega_o + \dot{\omega}(T_1 - t_o) \tag{19}$$

where
- ω_e ≡ the rotation rate of the earth and is a positive value (rad/sec)
- t_o ≡ the time of injection

The time of perigee passage for the first rev is obtained from

$$\tau_1 = T + (k+1)(2\pi/n) \tag{20}$$

where

T ≡ the input time of perigee passage

k ≡ value from equation (13)

The right ascension of the node at the time of nodal crossing is given by

$$\lambda_1 = [\alpha_{g_o} + \omega_e T_1 + \Omega_1]_{MOD\ 2\pi} \tag{21}$$

where

α_{g_o} ≡ the right ascension of Greenwich at midnight from the subroutine LFPA

The angle between the sun and the orbit plane will be evaluated for the first rev of each day by differencing the right ascension of the ascending node and the right ascension of the sun. The right ascension of the sun employs many of the same equations just presented except the sun is the central body and the earth is the "vehicle". These calculations are performed by LFPA. From Equations (5) and (6) we obtain

$$M = n(\Delta t) = \frac{2\pi}{365.25}(\Delta t)$$

where Δt ≡ the number of days since perihelion passage

and $M = E_s - e_E \sin E_s$

where E_s ≡ eccentric anomaly of the earth about the sun

e_E ≡ eccentricity of the earth's orbit

The true anomaly of the earth in its orbit about the sun is obtained from equation (4) which is

$$\nu = \sin^{-1}\left[\frac{\sqrt{1-e_E^2}\ \sin E_s}{1-e_E \cos E_s}\right]$$

The right ascension of the sun is obtained from

$$\gamma = \tan^{-1}\ [\cos i_S \tan (\nu - V)]$$

where γ ≡ right ascension of the sun

i_S ≡ obliquity of the ecliptic

V ≡ true anomaly of the earth at vernal equinox

The angle between the sun and the orbit plane is the difference of right ascensions which is

$$\beta = [\gamma - \lambda_1]_{\text{MOD } 2\pi}$$

where λ_1 ≡ right ascension of the node from equation (21)

The rate of change of the angle between the sun and the orbit plane is obtained by differencing the values of β for that day and the preceding day and dividing by the time difference

$$\dot{\beta} = (\beta_i - \beta_{i-1})/(t_i - t_{i-1})$$

where t_i ≡ time of the ascending node for the first rev of the ith day.

For the initial day of the run the value of β_{i-1} will have to be computed for the time, t_{i-1}. The time t_{i-1} will be obtained for the first day by subtracting 24 hours from t_i.

The values of a_1, e_1, and i_1 for the first rev are the same as the input values. This completes the set of orbital parameters which will be considered by the HECTOR routine. The complete set of these parameters is

a_1 ≡ semi major axis

e_1 ≡ eccentricity

i_1 ≡ inclination

Ω_1 ≡ longitude of the ascending node at the time of nodal crossing

ω_1 ≡ argument of perigee at the time of nodal crossing

τ_1 ≡ time of perigee passage

λ_1 ≡ right ascension of the node at the time of nodal crossing

T_s ≡ time of the start of the rev (i.e., the southern-most point on the rev)

β ≡ the angle in the equatorial plane from the ascending node of the vehicle to the sun (orbit plane sun angle)

$\dot{\beta}$ ≡ rate of change of the orbit plane sun angle

3.2.3 Computation of the Elements for Succeeding Revs

When the elements for the j^{th} rev of the day has been computed the elements of the $j + 1^{st}$ rev are obtained from the following:

$$a_{j+1} = a_j \tag{22}$$

$$i_{j+1} = i_j \tag{23}$$

$$e_{j+1} = e_j \tag{24}$$

$$\Omega_{j+1} = [\Omega_j + (\dot{\Omega} - \omega_e)(2\pi/n)]_{MOD\ 2\pi} \tag{25}$$

$$\omega_{j+1} = [\omega_j + \dot{\omega}(2\pi/n)]_{MOD\ 2\pi} \tag{26}$$

$$\tau_{j+1} = \tau_j + 2\pi/n \qquad (27)$$

$$T_{s_{j+1}} = T_{s_j} + 2\pi/n \qquad (28)$$

It should be noted that T_s is the time of the start of the rev in seconds from midnight and if $T_{s_{j+1}}$ is greater than 86,400 seconds the j^{th} rev was the last rev of the day. The first rev of the next day will be computed in the same manner as just described with the exception of time. For the start of a day, other than the day of launch, the time of perigee passage and the time of the start of the rev will be computed as:

$$\tau_1 = \tau_j + \frac{2\pi + (\dot{\omega} * \Delta t)}{n} - 86,400 \qquad (29)$$

$$T_{s_1} = T_{s_j} + 2\pi/n - 86,400 \qquad (30)$$

where

τ_j ≡ time of perigee passage of the last rev of the preceeding day

T_{s_j} ≡ time of the start of the last rev of the preceeding day

Δt ≡ $(86,400 - t_o)$ for the second day where t_o is time of injection and

≡ 86,400 for all days greater than the second

3.2.4 Compute Ground Track and Swath Dates for One Rev

The subroutine SWATH will be called to generate a table of visibility swath information for a specified vehicle. The time of perigee passage (TP_{VEH}), mean motion (n_{VEH}) and the eccentric anomaly at the ascending node (E_{Ω}) will have been computed by the calling routine and available in common. For one rev, the subroutine SWATH will compute vehicle position data at fixed increments of eccentric anomaly. For each vehicle position, the swath data in terms of latitude and longitudinal difference from the node will be computed for a series of cross-track scan angles.

The vehicle position data computations for the ith point in the orbit is initiated by the relationship:

$$E = E_{\Omega} + \Delta E * (i-1) \tag{31}$$

where,

ΔE = increment for orbit

Then, $T = (E - e \sin E)/n_{VEH} + TP_{VEH}$

where,

e = orbit eccentricity for the vehicle from the ephermeris data.

The sine and cosine of the true anomaly (υ), the argument of latitude (ω_T) and the vehicle altitude (H) can be computed by:

$$\sin(\nu) = \sqrt{1 - e^2} \quad \sin E/(1-e \cos E) \tag{32}$$

$$\cos(\nu) = (\cos E-e)/(1-e \cos E) \tag{33}$$

$$\omega_T = \nu + \omega_P \tag{34}$$

$$H = ER - SM\,(1-e^2)/(1+e \cos \nu) \tag{35}$$

where

ω_P = argument of perigee for the vehicle from the ephemeris data

SM = semi-major axis for the vehicle from the ephemeris data

ER = radius of the earth in the units of SM

The vehicle geocentric latitude is

$$Z_V = \sin \omega_T * \sin IN \tag{36}$$

$$\lambda_V = ARSIN\ (Z_V) \tag{37}$$

where

IN = orbit inclination for the vehicle from the ephemeris data

The right ascension of Greenwich and the longitude of the node at the time of the ascending node for the point being computed can be obtained from the relationships given by Equation (21).

The cartesian components of the instantaneous node vector in earth centered rotating (ECR) coordinates are:

$$X_N = \cos \phi_\Omega \tag{38}$$

$$Y_N = \sin \phi_\Omega \tag{39}$$

The vehicle position longitude and ECR components can be obtained from:

$$\phi_V = \phi_\Omega + ARTAN\ (\tan \omega_T \cos IN) + K * \pi \qquad \text{(Radians)} \tag{40}$$

$$X_V = \cos \lambda_V \cos \phi_V \tag{41}$$

$$Y_V = \cos \lambda_V \sin \phi_V \tag{42}$$

where $K = 1$ if $90^\circ \leq \omega_T \leq 270^\circ$, otherwise $K = 0$.

The vectors $\overline{N}$ and $\overline{Z}_B$ are defined to be:

$$\overline{Z}_B = (-V_X)i + (-Y_V)j + (-Z_V)k \tag{43}$$

$$\overline{N} = (X_N)i + (Y_N)j \tag{44}$$

where i, j, k are the ECR coordinate axes directions. The unit vector $\overline{Y}_B$ is then defined to be

$$\overline{Y}_B = (\overline{N} \times \overline{Z}_B) / \left| \overline{N} \times \overline{Z}_B \right| \tag{45}$$

The vector $\overline{P}_V$ is defined to be

$$\overline{P}_V = (D * X_V)i + (D * Y_V)j + (D * Z_V)k \tag{46}$$

where $D = 1 + H$ with H in earth radii.

The visibility swath data can be obtained for the vehicle position by computation of the latitude and longitude from the node of the intersection of the earth and a series of rays from the vehicle at various cross-track scan angles. For the j^{th} scan angle ($j = 1, 2, 3$)

$$L_Y = \sin (SA_j) \tag{47}$$

$$L_Z = \cos (SA_j) \tag{48}$$

$$\overline{L} = (A)(M) \tag{49}$$

where A is the two element row matrix (L_Y, L_Z)

M is the 2x3 matrix $\begin{pmatrix} \overline{Y}_B \\ \overline{Z}_B \end{pmatrix}$

$$S = -B - \sqrt{B^2 - H}$$

where $B = \left(\overline{L} \cdot \overline{P}_V\right) / |\overline{L}|$

and H is in earth radii.

If the value of S is negative, an intersection of the vector at that scan angle does not occur. If S is positive:

$$\overline{R} = \overline{P}_V + \frac{S\,\overline{L}}{|\overline{L}|} = (R_1)i + (R_2)j + (R_3)k \tag{50}$$

$$\lambda_j = \text{ARSIN } (R_3) \tag{51}$$

$$\phi_j = \text{ATAN 2 } (R_2, R_1) - \phi_\Omega \tag{52}$$

$$\tau_j = T - TA \tag{53}$$

These relationships yield the latitude, longitudinal difference and time relative to the node for the scan angles. A table containing these scan angle data along with the vehicle altitude, H, for each vehicle position data point may be output at completion of the processing and is written on the swath file.

3.2.5 Compile Nodal Search Data

The parameters needed for potential access determination in SAGE are:

$$\Omega_{MIN} = MIN\ (\Omega_1,\ \Omega_2) \tag{54}$$

$$\Omega_{MAX} = MAX\ (\Omega_1,\ \Omega_2) \tag{55}$$

where

$$\Omega_1 = 2\pi + \phi_3 - 2\phi_2$$

$$\Omega_2 = 2\pi + \phi_1 - 2\phi_1$$

and

ϕ_j obtained from Equation (52).

The Ω_{MIN}, Ω_{MAX} parameters are a function of latitude and the data represents a conservative estimate for nodal bounds for a potential access condition.

REVTAB receives through common a time ordered table defining the vehicle position and the visibility swath. For each time point the following data is provided in the table

- Time from the ascending node for the ith entry
- Vehicle altitude
- Vehicle latitude
- Longitudinal difference between the longitude of the node at the time of the node and longitude of the vehicle subpoint
- Swath data for each of scan angles
  - Latitude of scan angle vector intersection with the earth
  - Longitudinal difference between the node and the intersection of the scan vector with the earth

REVTAB will reformat the above data to provide the information in terms of latitude rather than time. The initial effort of REVTAB in generating the data for the table is to interpolate and compute the vehicle altitude, vehicle longitudinal difference and the time from the ascending node for each integer degree value of vehicle latitude. Since the vehicle crosses each latitude twice, once ascending and once descending, two sets of data would be needed to define both day and night passage. However, since a sun synchronous orbit is given only descending, passage is processed.

The swath data is processed in the same general manner. For each of the scan angles, the longitudinal difference between time from the ascending node are obtained by interpolation for each integer degree of scan angle vector latitude. The minimum and maximum latitudes for which table values can be obtained are the latitudes equal to the orbit inclination, IN (or 180° - IN if IN > 90°). For latitude values outside the range $\pm$ IN, the table entries will be zero to insure the proper subsequent processing of the data.

4.0 OUTPUT

4.1 GENERAL

There are five media for output. These are:

1. Intermediate debug print for swath calculations
2. Ephemeris display report
3. Echo print of card images
4. Swath table file
5. Swath reference file

All debug output from HECTOR will be deleted.

4.2 INTERMEDIATE DEBUG PRINT

The following debug prints occur if the input flag LIDEBG(1) is set to .TRUE.

1. For each vehicle and day

   SWATH OUTPUT - LAT. AND DELTA LONG FOR 3 POINTS, ALT AND TIME FOR LATITUDE POINT
   XXX.XXXXXXX XXX.XXXXXXX ⟶ for 6 more values XXXX

The following debug print occurs if LIDEBG(4) is .TRUE.

1. For each vehicle and day:

   MXLAT, MNMX, MXMN, NNLAT=XXXX XXXX XXXX XXXX

   For each of 3 scan angles:

   SLAM(361, L)=XX.XXXXXXXX ⟶ 2 more values
   SLAM(362, L)=XX.XXXXXXXX ⟶ 2 more values

2. The following debug print is produced for each lat. above minimum of swath edges at North Pole, for each lat. from lowest swath edge at North Pole to highest swath edge at South Pole and for each lat. from highest swath edge at South Pole to lowest swath edge at South Pole.

   For each vehicle and day and each of 3 scan angles, these data are printed:

   LATITUDE, TIME, and DLONG.

3. For each vehicle and day and for each of 3 scan angles, these data are printed:

   REVTAB OUTPUT - TIME AND DELTA LONG FOR SCAN ANGLES AND ALT

   TIME, DLONG, ALT and LATITUDE

The following debug print of swath file records occurs if L1DEBG(2) is .TRUE. and if the day is within one of the PRTI5 print selection pairs.

1. A dump of each record on the swath reference and swath table is given which meets the above criteria. The records would be printed in pairs as follows:

   SWATH REFERENCE RECORD   VEHICLE n   DAY nnn

WP	NREV	TMNODE	CARUS
±X. XXXXXXE±YY	1	nnnnn	±X. XXXXXXE±YY
	2	↓	↓
	|		
	17		

   SWATH TABLE RECORD   VEHICLE n   DAY nnn

LATNO	IALT	TIME(1)	TIME(2)	TIME(3)	DLONG(1)	DLONG(2)	DLONG(3)
65	nnnn	nnnnn	nnnnn	nnnnn	±X. XXXXXXE±YY ⟶		
64	↓	↓	↓	↓	↓	↓	↓
|							
-65							

4.3 REPORTS

There is only one report which will be printed and only if there are non-zero entries in IPPI5 input data array. The particular days that will be printed are under control of this array.

FORMAT:

```
              EPHEMERIS DISPLAY FOR DAY NNN

                       VEHICLE N

SEMI-MAJOR AXIS          =   XXXXXX.XXX
ECCENTRICITY             =   XX.XXXXXXXX
PERIOD                   =   XXXXXX.XXX

INCLINATION              =   XXXX  XX  XX
ARGUMENT OF PERIGEE      =   XXXX  XX  XX
RT. ASC. GREENWICH       =   XXXX  XX  XX

REV     NODE TIME            NODE LONGITUDE
 1)   XXXXXXX.XXXX           XXXX  XX  XX
 2)          |                    |
 |           |                    |
 v           v                    v
17)   XXXXXXX.XXXX           XXXX  XX  XX
```

4.4 DATA FILES

There are two output files. These are the swath table and the swath reference. See Section 2.4 of the Users Manual for a description of these files.

Record Length: 80 bytes or 55 words

Block Factor: 4

File Size: 197,820 bytes or 60,445 words

Usage: This file is used in conjunction with the swath table by SAGE to determine when and if a segment is accessed. The swath reference file is searched for a day and rev entry within a supplied delta of a given nodal longitude and time. The day obtained in this manner is used to read the desired record from the swath table. The calculations on the data can then be performed to obtain a further check for access.

4.5 ECHO PRINT OF INPUT CARD IMAGES

The control card input to EPHEM is printed after all nominal values have been stored.

HEADER

$AAAA \longrightarrow AAA$

ICASE	IVEH	NODAY	INLAT	ISLAT	LIDEBG	IPPIS	START/STOP	ISOSTR
NNNN	NN	NNN	NN NN	NN NN	LLLL	$NNN_1 NNN_2$----	NNN_{10}	NN_{01}

SA — IVITIM — IV2TIM

NNN.N NNN.N NNN.N NN NN NN/NN NN NN NN NN NN/NN NN NN 02

ORBIT 1 — ORBIT 2

NNNN.N NNNN.N NNN. NNN. NNN. NNNN.N NNNN.N NNN. NNN. NNN. 03

5.0 ERROR PROCESSING

5.1 GENERAL

The program will attempt to find as many sources of error during the input card processing as possible. The program will continue checking for further input errors upon detecting any input error. Since most of the computations are contained in subroutines which already exist, the philosophy of continuing processing after detection will be retained. There are two levels of error. These are:

Level 1 - continue processing

Level 2 - job fatal

When a level 1 error occurs, the program will print an informative message and continue. If such an error occurs during the calculation phase, a printout of key data quantities is given in addition. When a level 2 error occurs, the program will print an informative message and return control back to the computer system.

Level 2

1. A check is made to see if NODAY is between 1 and 549. Message:

   ***NODAY IS OUT OF RANGE. IT IS NOT BETWEEN 1 AND 549

2. A check is made to see if INLAT and ISLAT are each between 0 and 65. In addition, ISLAT(1) (INLAT(1)) must be ≤ ISLAT(2), (INLAT(2)). Message:

   ***INLAT OR ISLAT IS OUT OF RANGE. THEY ARE NOT BETWEEN 0 AND 65 OR THE FIRST VALUE OF EITHER ONE IS NOT LESS THAN OR = TO THE SECOND.

3. A check is made to make sure the number of latitude points is less than 101 as specified by ISLAT and INLAT. Message:

***THE TOTAL NUMBER OF LATITUDE POINTS
SPECIFIED BY ISLAT AND INLAT EXCEEDS 100.

4. A check is made to see if each entry pair for IPPI5 has entries between 1 and 549 and the first entry of a pair is ≤ to the second. Message:

   ***IPPI5(N) AND IPPI5(N+1) ARE OUT OF RANGE. THEY
   ARE NOT BETWEEN 1 AND 549 OR THE VALUE OF
   THE FIRST ENTRY IN A PAIR IS NOT LESS THAN OR =
   TO THE SECOND.

5. IVEH is checked to be between 1 and 2. No more than 2 vehicles can be processed by this program. Message:

   ***IVEH IS NOT BETWEEN 1 AND 2.

6. A check is made to see if each SA entry is in the range of -10 to +10 and is in ascending order and unequal to each other. Message:

   ***EITHER AN SA ENTRY IS NOT BETWEEN 0 AND
   ABS(10) OR ENTRY(N) IS NOT LESS THAN ENTRY(N+1)

7. A check is made to make sure IV1TIM(1) or IV2TIM(1) is greater than 1963. Message:

   ***EITHER IV1TIM(1) OR IV2TIM(1) IS NOT GREATER THAN 1963.

8. ORBIT1(1) and ORBIT2(1)-a are checked to be in the range of 6650 and 7700 kilometers. Message:

   ***EITHER ORBIT1(1) OR ORBIT2(1) IS NOT BETWEEN
   6650 AND 7700 KILOMETERS.

9. A check is made to make certain that ORBIT1(2) and ORBIT2(2)-e is in the range .00001 and .15. Message:

   ***EITHER ORBIT1(2) OR ORBIT2(2) IS NOT BETWEEN
   .00001 AND .15.

10. A check is made to make sure that ORBIT1(4) and ORBIT2(4)-Ω is in the range 0 and 360. Message:

    ***EITHER ORBIT1(4) OR ORBIT2(4) IS NOT BETWEEN 0 AND 360.

11. A check is made to make sure that ORBIT1(5) and ORBIT2(5)-ω is in the range 0 and 360. Message:

    ***EITHER ORBIT1(5) OR ORBIT2(5) IS NOT BETWEEN 0 AND 360.

12. A check is made to make sure that ORBIT1(6) and ORBIT2(6)-τ is in the range ± 65. Message:

    ***EITHER ORBIT1(6) OR ORBIT2(6) IS NOT BETWEEN -65 AND +65.

13. A check is made to make sure IV1TIM(4) or IV2TIM(4) specifies an hour such that the local vehicle passage time is between 700 to 1700 hours. Message:
    ***IV1TIM(4) OR IV2TIM(4) SPECIFIES A LOCAL VEHICLE PASSAGE TIME NOT BETWEEN 700 AND 1700 HOURS.

14. A check is made to make sure IV1TIM(3) or IV2TIM(3) is in the range 1 and 31. Message:

    ***IV1TIM(3) OR IV2TIM(3) IS NOT BETWEEN 1 AND 31.

15. A check is made to make sure IV1TIM(2) or IV2TIM(2) is in the range 1 and 12. Message:

    ***IV1TIM(2) OR IV2TIM(2) IS NOT BETWEEN 1 AND 12.

16. If one or more of the above errors occur, the following message will print:

    ***THIS JOB IS ABANDONED DUE TO THE FACT THAT 1 OR MORE FATAL INPUT ERRORS OCCURRED.

5.3 PROCESSING ERRORS

Level 1

1. A check is made in subroutine HECTOR after attempting to solve for the eccentricity anomaly that the iteration loop converged on a solution. Message:

   ***ERROR (KEPLER) SOLUTION FOR ECC. ANOMALY DID NOT CONVERGE AFTER 50 ITERATIONS.

2. A check is made in subroutine REVTAB to determine if an anomaly occurred in calculating crosstrack latitudes. If so, the following message is printed followed by data:

   REVTAB ERROR - ANOMALY IN CROSSTRACK LATITUDES

   Values for K, L, AA, JJ, DEL, XI, INT, NTRY, and LLL are then printed.

3. A check is made in subroutine SWATH to make sure the iteration count of 5 is not exceeded in computing geocentric latitude and delta longitude. Message:

   Iteration limit on SWATH. Values for XLONGN, TEM, DXLON, R(1), R(2), IFLG are then printed.

5.4 INPUT/OUTPUT ERRORS

For sequential I/O the FORTRAN system on the UNIVAC or PDP takes control and prints a message identifying the problem and will either continue processing or abandon the job. If processing continues, the system counts the number of times this error re-occurs and if it happens a certain number of times, the system will abandon the job.

For direct access I/O, the UNIVAC or PDP D.A. I/O package prints out an informative message, sets an error flag and allows processing to continue. In this program the swath table file is the only direct access file. Immediately after the informative message, the following message will print:

***AN IRRECOVERABLE I/O ERROR HAS OCCURRED ON WRITING A RECORD TO THE SWATH TABLE. THE JOB IS BEING ABANDONED.

PART II

COMMON BLOCK DEFINITIONS

COMMON STORAGE ALLOCATION

Name CONS Size

Page of

Function Contains constants and input data

Name	Dimension	Format	Description	Symbol	Units
WE	1	R	.72921151E-4		
NVEH	1	I	No. of vehicles 2		
NODAY	1	I	549 no. of days in the run		
IVITIM	6	I	Vehicle 1 injection time no. nominal		
IV2TIM	6	I	Vehicle 2 injection time no. nominal		
ORBIT1	6	R	No nominals orbital elements vehicle 1		
ORBIT2	6	R	No nominals orbital elements vehicle 2		
KO	1	I	Output print unit = 6		
PI	1	R	3.1415926	π	
CJ2	1	R	.10823E-2		
CMU	1	R	.15362204E-5		
PI2	1	R	1.5707963	$\pi/2$	
PI32	1	R	4.712389	$3/2\pi$	
IORBIT	1	I	Orbit option set to 2		
SA	3	R	Scan reference angle no nominal		
LIDEBG	4	L	Debug flags nominal is false		
RADIAN	1	R	57.29578		

COMMON STORAGE ALLOCATION

Name CONS Size ________ Page 2 of ______

Function ____________________

Name	Dimension	Format	Description	Symbol	Units
ECCE	1	R	.167263E-1		
CIS	1	R	.409280		
CNUV	1	R	1.35695		
RADIUS	1	R	Radius of earth 6376.436		KILOM.
HEADER	20	R	Header for printing initialized to blanks		
ICASE	1	I	Case no. no nominal		
IPPIS	10	I	Report day print range nominal 1, 549, 8*0		
NAME	2	R	'SWATHbTB'		
NAME1	2	R	'SWATHbREF'		
NLINE	1	I	Line count for print set to 99		
NPAGE	1	I	Page no. set to 0		
TWOPI	1	R	6.2831852	2π	
ALPHAM	1	R			
THES	1	R			
INP	1	I	Card input unit = 5		
LINMAX	1	I	Max no. of lines per page set to 39		
ISWTB	1	I	8-swath table file unit		
ISWR	1	I	9-swath ref. file unit		

COMMON STORAGE ALLOCATION

Name FLAG Size ______ Page _____ of _____

Function ______

Name	Dimension	Format	Description	Symbol	Units
IFIRST	1	I	First time flag, ≠ 0 - first time, = 0 - not first time		
IVEH	1	I	No. of vehicles = 2 max		
ISTART	1	I	Case start date in days since 1950		
IEND	1	I	Case end date in days since 1950		
IDEND	1	I	Current noday for a given vehicle		

COMMON STORAGE ALLOCATION

Name SWATH Size 407 words Page ____ of ____

Function ____________________

Name	Dimension	Format	Description	Symbol	Units
INLAT	2	I	North latitude study band definition		DEG
ISLAT	2	I	South latitude study band definition		DEG
ISOSTR	1	I	Δ with respect to north start for south latitude start		MONTH
NLAT	1	I	No. of latitudes (N and S) to be studied		
LATNO	100	I	List of latitudes from North to South for swath		DEG
IALT	100	I	From alt list of corresponding altitudes with LATNO		KILOM * 10
IGRDI	17	I	For FOC ground sta. ind.		
DLONMN	100	I	Minimum Δlongitude table		RAD
DLONMX	100	I	Maximum Δ longitude table		RAD
TIME1	100x3	R	Data from time ready for writing - matches LATNO		
DLONG1	100x3	R	Data from DLONG ready for writing matches LATNO		

COMMON STORAGE ALLOCATION

Name NREV Size______ Page____ of____

Function______

Name	Dimension	Format	Description	Symbol	Units
NREV	2	-	Count of rev's for each vehicle		
NREVCT	1x2	I	Equivalent to NREV (1)		
XLONG	1x2	R			

COMMON STORAGE ALLOCATION

Name ACCEPH Size 47 words Page ____ of ____

Function ____

Name	Dimension	Format	Description	Symbol	Units
TMNODE	17x2	R	Time of crossing for ascending node		SEC
SM	1x2	R			
EC	1, 2	R			
XIN	1, 2	R			
RTASC	1, 2	R			
WP	1, 2	R			
TNDE	1, 2	R			
RG	1	R			

COMMON STORAGE ALLOCATION

Name ACCDAT Size 2898 words Page ____ of ____

Function Contains swath data

Name	Dimension	Format	Description	Symbol	Units
TV	362	R			
HV	362	R			
SLAM	362x3	R			
SPHI	362x3	R			
ENP	2	R			

COMMON STORAGE ALLOCATION

Name REVSTH Size 2534 words Page ____ of ____

Function ______________________

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
TIME	181x3x2	R			
DLONG	181x3x2	R			
ALT	181x2	R			

PART III

LIST OF SUBROUTINES

AND

SUBROUTINE CALL STRUCTURE

LIST OF ROUTINES IN EPHEM

NAME	FUNCTION
1. DAY	Computes IDAY - DAY no. of the year
2. PIMOD	Converts an angle in radians to be between 0 - 2π
3. LFPA	Computes right ascension and ZULU day
4. CLDAY	Computes month and day of current year
5. PAGER	Restores pages and prints header
6. KEPLER	Computes eccentric anomaly
7. ACCGEN	Controls calculation of the orbital swath
8. HECTOR	Compute ephemeris data
9. REVTAB	Generates swath data vs latitude points
10. SWATH	Computes swath table
11. ALPHA	Computes sun angle
12. DEGMOD	Converts an angle in radians to degrees
13. INPROC	Reads and checks input data control cards
14. LATSEL	Forms the latitude band for swath table output
15. WRITSW	Writes a record on the swath table and swath reference file.
16. PRTI5	Print the ephemeris report
17. MAIN	Main control routine for the program
18. ERRDET	Process's errors and prints messages

EPHEM SUBROUTINE CALL STRUCTURE

- MAIN
    - INPROC
        - EJECT
        - ERRDET
            - PAGER
    - LFPA
    - DAY
    - PAGER
    - HECTOR
        - CLDAY
        - LFPA
        - PAGER
        - KEPLER
        - ALPHA
        - PIMO0
    - WRITSW
    - ACCGEN
        - SWATH
            - PIMOD
            - PAGER
            - EJECT
        - REVTAB
            - PAGER
        - LATSEL
            - PIMOD
        - WRITSW
    - PRTI5 PRTI5
        - DEGMOD
        - EJECT
        - PAGER
    - EJECT

PART IV

SUBROUTINE DESCRIPTIONS

SUBROUTINE KEPLER

Purpose:

This routine solves KEPLERS equations for eccentric anomaly (XE), given eccentricity (XECC) and mean anomaly (XM). The equations for the elliptic case is XM = -XECC* & IN(XE).

Calling Sequence:

Call KEPLER (XM, XECC, XE, ERROR)

SUBROUTINE ACCGEN

Purpose:

This routine initializes data for access's, computes preliminary orbit parameters, computes the eccentric anomally at ascending node and time of perigee passage. It also controls the calculation of the orbital swath and latitude interpolation tables.

SUBROUTINE HECTOR

Purpose:

This routine generates the ephemeris data for two vehicles for one day. The ephemeris is maintained as a set of orbital elements.

SUBROUTINE REVTAB

Purpose:

This routine receives as input the swath data table from the subroutine swath. REVTAB reformats the table to obtain the data at each integer value of latitude.

SUBROUTINE SWATH

Purpose:

This routine utilizes the ephemeris data for the day specified to generate the latitude and delta longitude data for the scan angles for one rev. The data is generated at one degree increments of eccentric anomaly, beginning at the ascending node.

Subroutine INPROC

ENTRY POINT: Call INPROC

PURPOSE: This routine reads the job header card, the program control card and 2 data cards as discussed in the problem description document. It also calls ERRDET to find input errors then performs initial calculations.

INPUT: All data in the input data cards as discussed in Section 2.0 - Problem Description.

OUTPUT: CONS COMMON

ICASE, NODAY, NVEH, HEADER, LIDEBG, IV1TIM, IV2TIM, ORBIT1, ORBIT2, SA, INP

FLAG COMMON

IVEH

SWAT COMMON

INLAT, ISLAT, IPPIS, ISOSTR, LATNO

SUBROUTINES USED:

CALL ERRDET

METHOD/PROCEDURE

- Read NODAY, IVEH, INLAT, ISLAT and IPPI5 into a temporary storage area and the rest of the data directly into working storage. Move the above items into working storage only if ≠ 0. Call ERRDET to check input for errors. Perform initial calculations per Sec. 3.4.3. and in last paragraph item 3.

Subroutine LATSEL

ENTRY POINT: Call LATSEL

PURPOSE: To move swath data from a full 180° swath table storage to an area for writing on the swath table. This routine also computes the minimum and maximum Δ longitude for all required latitudes.

INPUT: SWAT COMMON
NLAT, LATNO, DLONMX, DLONMN
REVSTH COMMON
TIME, DLONG, ALT
CONS COMMON
TWOPI

OUTPUT: SWAT COMMON
IALT, DLONMN, DLONMX, TIME1, DLONG1

SUBROUTINES USED: None

METHOD/PROCEDURE

See the 4th and 5th paragraphs of Section 3.4 item 4.

Subroutine WRITSW

ENTRY POINT: Call WRITSW (IOPT)

PURPOSE: To initialize, write and terminate output records on the swath table file - ISWTB and swath reference file - ISWR. A record is written for each vehicle and file.

INPUT: CONS COMMON

NAME, NAME1, ICASE, NVEH, NODAY, IV1TIME, IV2TIM, ISWTB, ISWR

CALLING SEQUENCE: IOPT

HECCOM COMMON

XOMEGD(2) - 21^{th} LOC, XWD - 19^{th} LOC

ACCNUM COMMON

KDAY, KVEH

ACCEPH COMMON

WP (1, KVEH), TMNODE (i, KVEH), i = 1, 17

NREV COMMON

NREV (1, KVEH)

ICARUS COMMON

(CARUS (i, 1, j), i = 1, 3, j = 1, 2), (CARUS (4, i, KEVH), i = 1 - 17)

SWAT COMMON

INLAT, ISLAT, ISOSTR, (LATNO (i), IALT(i), (TIME1 (i, n), DLONG1(i, n), n = 1, 3), DLONMX(i), DLONMN(i), i = 1, NLAT)

OUTPUT: See swath table and swath reference table output formats in Problem Description document

DACDC COMMON: INDEXR

SUBROUTINES USED:

FOR CDC - OPENMS, READMS, WRITMS, SLOSEMS

D. A. PACKAGE

FOR UNIVAC - The define file PACKAGE

METHOD/PROCEDURE

The routine functions according to the value in IOPT

For IOPT = 1 - initialization

SWATH TABLE

a. A define file is given to initialize pointers and open the file.

b. Data is moved from common storage to the output array A(800).

c. The header record is written.

SWATH REFERENCE - Rec. No. 1

a. A rewind is given.

b. Header record written.

For IOPT = 2 - write detail records

SWATH TABLE

a. Data for detail record is moved into "A"

b. Write detail record, get rec. no. from formula in Problem Definition.

SWATH REFERENCE

a. Write detail record

For IOPT = 3 - write swath table trailer record

a. Data for trailer record is moved into "A"

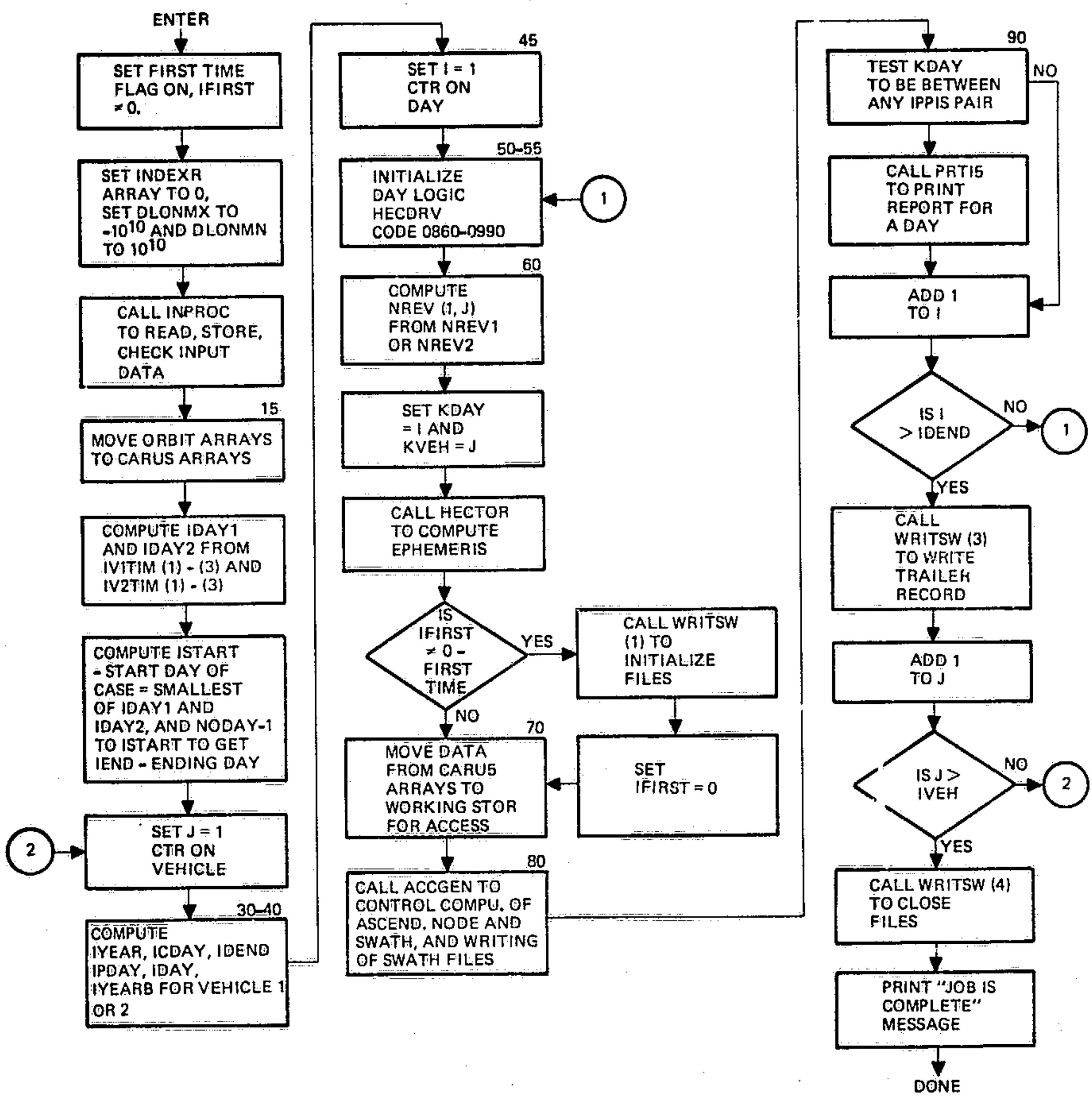
b. Write the record, rec. no. is obtained from formula in Problem Definition

For IOPT = 4 - close files

SWATH TABLE - Automatic close

SWATH REFERENCE - Write EOF and rewind.

SUBROUTINE MAIN BLOCK DIAGRAM



Subroutine ERRDET

ENTRY POINT: Call ERRDET

PURPOSE: This routine performs all input error checking for the data on cards. It will continue to check for all errors even though 1 or more errors were detected. Having found at least one error the program will stop.

INPUT: CONS COMMON

NODAY, SA, IV1TIM, ORBIT1, ORBIT2

SWAT COMMON

INLAT, ISLAT, IPPIS, S

FLAG COMMON

IVEH

OUTPUT: Error messages as shown in Section 5. 2

METHOD/PROCEDURE

When an error occurs a counter is updated. When the last error has been checked this counter is tested if = 0 then a normal return is executed, otherwise a termination message is printed and a stop is executed. The errors to be checked are found in Section 5. 2.

SUPPLIED UTILITY ROUTINES

Routine Day

Call Day (IYMD, IDAY)

Given IYMD (3) where	IYMD (1)	IS Day No.
	IYMD (2)	IS Month No.
	IYMD (3)	IS Year No.

Compute year day no. in IDAY

Routine PIMOD

Call PIMOD (A)

Convert ±A in radians to an angle $0-2\pi$

Routine SOL (Entry ALPHA)

Call ALPHA (IFLAG)

For emphemeris usage as called by hector

computes ALPHAM and ALPHAT and IFLAG = 1

Routine PAGER (Entry Eject)

Call PAGER (NLINES)

Updates line count in NLINE with NLINES

NPAGE = 0 causes page to be restored prior to print.

NPAGE - page no.

HEADER- 80 char. 20A5

ICASE- case no.

KO - 6 print unit

INMAX is max no. of lines allowed

Initially NLINE should be set > LINMAX and NPAGE = 0

SUPPLIED UTILITY ROUTINES
(CONTINUED)

Call EJECT (NLINES)

Causes page to be restored automatically and then prints headers.

Routine CLDAY

Call CLDAY

Given IDAY-DAY no. of the year compute in LMO-the month and in LDA the day no.

Need: IYEAR = 0 - Leap Year, ≠ 0 not Leap Year

Routine KEPLER

Call KEPLER (XM, XECC, XE, ERROR)

Given XM - Mean anomaly, XECC - eccentricity

Compute: E-eccentric anomaly, error = 0 means OK

Routine LFPA

Call LFPA [FLDA, LMO, LYR, ALFGM (can be dummy), DAYS]

Given: FLDA - day of month no., LMO - month no.,
LYR - year no. compute ALFGM - right ascension and
DAYS - Zulu day no.

Routine DEGMOD

Call DEGMOND (RAD, IDEG)

Given: angle rad in radians store the angle in deg., min., sec., in IDEG(1) - (3).

Routine FZULU

Call FZULU (IOATE, IOUT)

Given Zulu date in IDATE, compute year, month and day in IOUT(1) - IOUT(3).

Routine RDMIA

Call RDMIA(FL, U)

Given double precision random no. seed in FL, comoute random no. U (0-1) based on uniform distribution.

PART V

SUBROUTINE LISTINGS

ORIGINAL PAGE IS OF POOR QUALITY

```
ACCGEN,1,751031, 57154    + 1

      SUBROUTINE ACCGEN                                                  ACCGEN
C                                                                        ACCGEN
C                                                                        ACCGEN
      COMMON /CONS/                                                      CONS
     1ECCF,EIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAN,     CONS
     2THES,FF,ORBTT1(6),ORBTT2(6),PI,CJ2,CNU,PI2,PI32,SA(3),RADIAN,      CONS
     3ICASE,IPPIS(10),NLINE,NPAGE,INP,LINMAX,ISWIH,ISNR,NVEH,NUDAY,      CONS
     4IV1TIM(6),IV2TIM(6),KD,IORBIT,LIDFBG(4)                            CONS
      REAL NAME,NAME1                                                    CONS
      LOGICAL LIDFBG                                                     CONS
C                                                                        ACCGEN
      DIMENSION MAXLT(2),MINLT(2)                                        ACCGEN
      DIMENSION DELNDE(2),PER(2)                                         ACCGEN
      DIMENSION TIMNDE(19,2),XLONND(19,2),MXREV(2)                       ACCGEN
      COMMON/ACCNUM/ACCNUM(100)                                          ACCGEN
      EQUIVALENCE (ACCNUM(  1),KVEH)                                     ACCGEN
     X,          (ACCNUM(  3),NOSCN)                                     ACCGEN
     X,          (ACCNUM(  4),NAUSCN)                                    ACCGEN
     X,          (ACCNUM(  6),XN )                                       ACCGEN
     X,          (ACCNUM(  7),PER(1))                                    ACCGEN
     X,          (ACCNUM(  9),TIMNDE(1,1))                               ACCGEN
      EQUIVALENCE (ACCNUM( 47),XLONND(1,1),ARRAY1(1))                    ACCGEN
     X,          (ACCNUM( 66),ARRAY2(1))                                 ACCGEN
     X,          (ACCNUM( 85),DELNDE(1))                                 ACCGEN
     X,          (ACCNUM( 87),MXREV(1))                                  ACCGEN
     X,          (ACCNUM( 89),KDAY)                                      ACCGEN
     X,          (ACCNUM( 90),E)                                         ACCGEN
     X,          (ACCNUM( 91),IPER)                                      ACCGEN
     X,          (ACCNUM( 92),MAXLT(1))                                  ACCGEN
     X,          (ACCNUM( 94),MINLT(1))                                  ACCGEN
     X,          (ACCNUM( 96),IADF)                                      ACCGEN
C                                                                        ACCGEN
C                                                                        ACCGEN
      COMMON /NPEV/                                                      NPEV
     1XLONG(1,2),NREV(2)                                                 NPEV
      DIMENSION NPEVCT(1,2)                                              NPEV
      EQUIVALENCE (NREV(1),NREVCT(1,1))                                  NPEV
      COMMON /ACCEPH/                                                    ACCEPH
     1TIMNDE(17,2),SM(1,2),FC(1,2),XIN(1,2),RTASC(1,2),WP(1,2),TNDE(1,2)ACCEPH
     2,PD(1)                                                             ACCEPH
      COMMON /REVSTH/                                                    REVSTH
     1TIME(181,3,2),DLONG(181,3,2),ALT(181,2)                            REVSTH
      COMMON /FLAG/                                                      FLAG
     1IFIRST,IVEH,ISTART,IEND,IDEND                                      FLAG
      COMMON /SRAT/                                                      SRAT
     1CLONMN(100),CLONMX(100),TIME1(100,3),PLONG1(100,3),INLAT(2),      SRAT
     2ISLAT(2),ISPSTR,NLAT,LATNO(100),IALT(100),ICRDI(17)                SRAT
C                                                                        ACCGEN
C     INITIALIZE DATA FOR ACCESSES                                       ACCGEN
C                                                                        ACCGEN
C                                                                        ACCGEN
      DO 8 I = 1,3                                                       ACCGEN
      IF ( SA(I) .EQ. 0.0) GO TO 8                                       ACCGEN
    5 NOSCN = 1                                                          ACCGEN
    8 CONTINUE                                                           ACCGEN
      NAUSCN = NOSCN/2 + 1                                               ACCGEN
      MULTIF = 0                                                         ACCGEN
      I = 1                                                  *NEW
```

```
      C                                                                   ACCGEN  **-1
            XLONG(I,1) = 1000000.                                         ACCGEN
            XLONG(I,2) = 1000000.                                         ACCGEN
      C                                                                   ACCGEN
      C  PROCESS THE ACCESS FOR EACH VEHICLE                              ACCGEN
      C                                                                   ACCGEN
      C                                                                   ACCGEN
      C  COMPUTE PRELIMINARY ORBIT PARAMETERS                             ACCGEN
            I= 1                                                          SET1
            J = KVEH                                                      SET1
      C                                                                   ACCGEN
            XN = SQRT(CMU/SM(I,J)**3)                                     ACCGEN
            XECC = EC(I,J)                                                ACCGEN
            PER(J) = TWOPI/XN                                             ACCGEN
      C                                                                   ACCGEN
      C  COMPUTE ECCENTRIC ANOMALY AT ASCENDING NODE                      ACCGEN
      C                                                                   ACCGEN
            TRM1 = (1.-XECC**2)/(1.+XECC*COS(-WP(I,J)))                   ACCGEN
            SINE = TRM1*SIN(-WP(I,J))/SQRT(1.-XECC**2)                    ACCGEN
            COSE = (1.-TRM1)/XECC                                         ACCGEN
            E =  ACOS(COSE)                                               ACCGEN
            IF ( SINE .LT. 0.0) E = TWOPI - E                             ACCGEN
      C                                                                   ACCGEN
      C  COMPUTE TIME OF PERIGEE PASSAGE                                  ACCGEN
      C                                                                   ACCGEN
            ENDE = E - TWOPI                                              ACCGEN
            TRM2 = (ENDE - XECC*SIN(ENDE))/XN                             ACCGEN
            TPER = TNDE(I,J) - TRM2                                       ACCGEN
            TEMP = TNDE(I,J) + PER(J)                                     ACCGEN
       15   IF( TPER .GE. TNDE(I,J) .AND. TPER .LE. TEMP) GO TO 80        ACCGEN
            IF ( TPER .GE. TNDE(I,J) ) GO TO 20                           ACCGEN
            TPER = TPER + PER(J)                                          ACCGEN
            GO TO 15                                                      ACCGEN
       20   TPER = TPER - PER(J)                                          ACCGEN
            GO TO 15                                                      ACCGEN
      C                                                                   ACCGEN
      C  GENERATE SWATH BOUNDARIES AND BUILD LATITUDE INTERPOLATION TABLE ACCGEN
      C                                                                   ACCGEN
      C                                                                   ACCGEN
       80   CONTINUE                                                      ACCGEN
            CALL SWATH                                                    ACCGEN
            CALL REVTAB                                                   ACCGEN
            CALL LATSEL                                                   ACCGEN
            CALL WRITSW(2)                                                ACCGEN
      C                                                                   ACCGEN
      C                                                                   ACCGEN
      100   CONTINUE                                                      ACCGEN
      C                                                                   ACCGEN
            RETURN                                                        ACCGEN
      C                                                                   ACCGEN
      C                                                                   ACCGEN
            END                                                           ACCGEN
```

```
@ FOR ALPHA,1,750925, 58800     , 1.
      SUBROUTINE ALPHA(IFLAG)
C                                                                 ALPHA
C                                                                 ALPHA
C     SUBROUTINE SOL PROVIDES THE SUN ELEVATION ANGLE OF AZIMUTH WHEN CALLEALPHA   **-4
C     FOR SPECIFIC TARGET                                         ALPHA
C                                                                 ALPHA
C                                                                 ALPHA
C              CALLING SEQUENCES...                               ALPHA
C                        CALL SOL(PHIT,THET,TM,ELEVS,AZIMS,IFLAG) ALPHA
C                                                                 ALPHA
C                             INPUT --  PHIT -- LONGITUDE OF TARGET  ALPHA
C                                       THET -- LATITUDE OF TARGET   ALPHA
C                                       TM   -- TIME                 ALPHA
C                                       IFLAG-- =0, COMPUTE SUN AZ/EL ALPHA
C                             OUTPUT -- ELEVS-- SUN ELEVATION ANGLE  ALPHA
C                                       AZIMS-- SUN AZIMUTH          ALPHA
C                        CALL ALPHA(IFLAG)                        ALPHA
C                                                                 ALPHA
C                             INPUT --  IFLAG-- =1 COMPUTE RT. ASC. OF TALPHA
C                                                                 ALPHA
      COMMON /CONS/                                               ALPHA
     1ECCF,CISTCNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM, CONS
     2THES,WE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,  CONS
     3ICASE,IPPI5(10),NLINE,NPAGE,INP,LINMAX,ISWTR,ISKR,NVEH,NUDAY,  CONS
     4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDFBG(4)                       CONS
      REAL NAME,NAME1                                             CONS
      LOGICAL LIDFBG                                              CONS
C                                                                 CONS
      COMMON /FLAG/                                               ALPHA
     1IFIRST,IVEH,ISTART,IEND,IDEND                               FLAG
      COMMON /HECCOM/                                             FLAG
     1ALPHAT,XBETAD(2),XBETA(2,2),XWD(2),XOMEGD(2),TASCNP(2),XI0(2), HECCOM
     2PERIOD(2),EMO,LDA,NREV1,NREV2,IYR,IYRB,IDAY,IPDAY,ICDAY,NREV1P, HECCOM
     3NREV2P,IV1DAY,IV2DAY,LYR                                    HECCOM
C                                                                 SETT
C                                                                 ALPHA
C     COMPUTE DAY OF YEAR (SINCE NOT AVAILABLE IN PREDICTOR)      ALPHA
C                                                                 ALPHA
C     ERROR TERM DUE TO VARIABLE SPEED OF ROTATION OF EARTH       ALPHA
         EP = 0.                                                  ALPHA
C     CALCULATE MEAN ANOMALY OF EARTH                             ALPHA
         DAY = IPDAY                                              ALPHA
         EM = (DAY-2.)*TWOPI/365.25                               ALPHA
C     CALCULATE ECC. ANOMALY OF EARTH                             ALPHA
         EP = .001/RADIAN                                         ALPHA
         E1 = EM                                                  ALPHA
   10 E2 = E1-((E1-EM-ECCF*SIN(E1))/(1.-ECCF*COS(E1)))            ALPHA
         ED = ABS(E2-E1)                                          ALPHA
         E1 = E2                                                  ALPHA
         IF(ED-EP) 20,20,10                                       ALPHA
   20 IF(E1-TWOPI) 40,50,30                                       ALPHA
   30 E1 = E1-TWOPI                                               ALPHA
   40 IF(ABS(E1-PI)-EP) 41,41,45                                  ALPHA
   45 IF(E1) 46,55,60                                             ALPHA
   46 E1 = E1+TWOPI                                               ALPHA
         GO TO 40                                                 ALPHA
C     TRUE ANOMALY OF EARTH                                       ALPHA
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
   41 XNUE = PI                                                          ALPHA
      GO TO 70                                                           ALPHA
   50 E1 = 0.                                                            ALPHA
   55 XNUE = 0.                                                          ALPHA
      GO TO 70                                                           ALPHA
   60 XNUE = 2.*ATAN(SQRT((1.+ECCE)/(1.-ECCE))*TAN(E1/2.))               ALPHA
      IF(XNUE) 65,70,70                                                  ALPHA
   65 XNUE = TWOPI+XNUE                                                  ALPHA
C     DECLINATION OF SUN                                                 ALPHA
   70 THES = ASIN(SIN(CTS)*SIN(XNUE-CNUV))                               ALPHA
      DIF = XNUE-CNUV                                                    ALPHA
      COSDIF = COS(DIF)                                                  ALPHA
      SINDIF = SIN(DIF)                                                  ALPHA
      IF(COSDIF) 90,75,90                                                ALPHA
   75 IF(SINDIF) 85,80,80                                                ALPHA
C     RT.ASC. OF TRUE SUN                                                ALPHA
   80 ALPHAT = PI2                                                       ALPHA
      GO TO 150                                                          ALPHA
   85 ALPHAT = 3.*PI/2.                                                  ALPHA
      GO TO 150                                                          ALPHA
   90 ALPHAT = ATAN(COS(CTS)*SINDIF/COSDIF)                              ALPHA
      IF(COSDIF) 105,95,95                                               ALPHA
   95 IF(ALPHAT) 100,150,150                                             ALPHA
  100 ALPHAT = TWOPI+ALPHAT                                              ALPHA
      GO TO 150                                                          ALPHA
  105 ALPHAT = ALPHAT+PI                                                 ALPHA
C     RT. ASC. OF MEAN SUN                                               ALPHA
  150 ALPHAM = (279.69668+360./365.25*DAY)/RADIAN                        ALPHA
C                                                                        ALPHA
 1000 RETURN                                                             ALPHA
      END                                                                ALPHA
```

@ ELT BLKDAT,1,750922, 69848 , 1

```
      BLOCK DATA                                                        BLOCK
      COMMON /SLAT/                                                     SWAT
     1DLONMN(100),DLONMX(100),TIME1(100,3),DLONG1(100,3),INLAT(2),      SWAT
     2ISLAT(2),ISOSTR,NLAT,LATNO(100),IALT(100),IGRDI(17)              SWAT
C     CONTAINS INITIAL VALUES FOR COMMON BLOCK QUANTITIES               BLOCK
      COMMON /CONS/                                                     CONS
     1ECCE,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,    CONS
     2THLS,WE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,     CONS
     3ICASE,IPPIS(10),NLINE,NPAGE,INP,LINMAX,ISWTH,ISWR,NVEH,NODAY,     CONS
     4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDEBG(4)                           CONS
      REAL NAME,NAME1                                                   CONS
      LOGICAL LIDEBG                                                    CONS
      COMMON /FLAG/                                                     FLAG
     1IFIRST,IVEH,ISTART,IEND,IDEND                                     FLAG
      DATA IVEH/2/                                                      BLOCK
      DATA WE/.72921151E-4/,NVEH/2/,NODAY/549/,IV1TIM/6*0./,IV2TIM/6*0/,BLOCK
     1ORBIT1/6*0./,ORBIT2/6*0./,KO/6/,PI/3.1415926/,CJ2/.10823E-2/      BLOCK
      DATA                                                              BLOCK
     1CMU/.15302204E-5/,PI2/1.5707963/,PI32/4.712389/,IORBIT/2/,SA/3*0./BLOCK
     2,LIDEBG/4*.FALSE./,RADIAN/57.29578/                               BLOCK
      DATA ECCE/.167263E-1/,CIS/.409280/,CNUV/1.35695/,RADIUS/6376.436/,BLOCK
     1HEADER/20*4H    /,IPPIS(1)/1/,IPPIS(2)/549/,IPPIS(3)/0/,          BLOCK
     2IPPIS(4)/0/                                                       BLOCK
      DATA NAME/4HSWAT,4HH TB/,NAME1/4HSWAT,4HH RF/,NLINE/99/,NPAGE/0/, BLOCK
     1TWOPI/6.2831852/,INP/5/,LINMAX/39/,ISWTH/8/,ISWR/9/               BLOCK
      DATA INLAT/5,65/,ISLAT/15,45/                                     SETT
      END                                                               BLOCK
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
 @ FLT CLDAY,1,751013, 57506      , 1

              SUBROUTINE CLDAY                                                  CLDAY
        C                                                                       CLDAY
        C MODULE - PPCLDAY                                                      CLDAY
        C OCTOBER 1, 1973                                                       CLDAY
        C                                                                       CLDAY
        C                                                                       CLDAY
        C  JUNE 26,1973                                                         CLDAY
        C                                                                       CLDAY
        C  MODULE PPCLDAY - POINT TARGET PERFORMANCE PREDICTOR                  CLDAY
        C                                                                       CLDAY
        C  GIVEN DAY OF YEAR(1-366), CLDAY COMPUTES THE MONTH AND DAY OF MONTH  CLDAY
        C                                                                       CLDAY
              COMMON /HECCOM/                                                   HECCOM
             1ALPHAT,XBETAD(2),XBETA(2,2),XWD(2),XOMEGD(2),TASCNP(2),XTO(2),    HECCOM
             2PERIOD(2),LMO,LDA,NREV1,NREV2,IYR,IYRB,IDAY,IPDAY,ICDAY,NREV1P,   HECCOM
             3NREV2P,IV1DAY,IV2DAY,LYR                                          SETI
              EQUIVALENCE (IYR,IYEAR)                                           CLDAY
              DIMENSION ITAB(13),ITAB1(13)                                      CLDAY
              DATA ITAB/1,32,60,91,121,152,182,213,244,274,305,335,             CLDAY
             *                    366/                                          CLDAY
              DATA ITAB1/1,32,61,92,122,153,183,214,245,275,306,336,            CLDAY
             *                    367/                                          CLDAY
        C  CHECK FOR LEAP YEAR                                                  CLDAY
        C                                                                       CLDAY
              IF(IYEAR.EQ.0) GO TO 20                                           CLDAY
        C                                                                       CLDAY
              DO 10 K=1,12                                                      CLDAY
              IF(ITAB(K).LE.IDAY .AND. IDAY.LT.ITAB(K+1)) GO TO 40              CLDAY
           10 CONTINUE                                                          CLDAY
              GO TO 40                                                          CLDAY
        C                                                                       CLDAY
        C  LEAP YEAR                                                            CLDAY
        C                                                                       CLDAY
           20 DO 30 K=1,12                                                      CLDAY
              IF(ITAB1(K) .LE. IDAY  .AND.  IDAY .LT. ITAB1(K+1)) GO TO 35
           30 CONTINUE                                                          CLDAY
         35   LMO = K
              LDA = IDAY - ITAB1(K) + 1                                                   *NEW
              GO TO 50                                                                    **-1
           40 LMO=K                                                             CLDAY
              LDA=IDAY-ITAB(K)+1                                                CLDAY
         50   RETURN
              END                                                               CLDAY
```

```
@ ECT DAY,1,750925, 58783        , 1

      SUBROUTINE DAY(IYMD,IDAY)                                    DAY
C                                                                  DAY
C                                                                  DAY   **-7
C  CONVERTS MONTH, DAY TO DAY NUMBER                               DAY
C                                                                  DAY
      DIMENSION IYMD(3),IMONTH(12),IMON(12)                        DAY
C                                                                  DAY
      DATA IMONTH/0,31,59,90,120,151,181,212,243,273,304,334/      DAY
      DATA IMON/0,31,60,91,121,152,182,213,244,274,305,335/        DAY
C                                                                  DAY
      MONTH=IYMD(2)                                                DAY
      IDAY=IYMD(3)                                                 DAY
      IYEAR=MOD(IYMD(1),4)                                         DAY
      IF(IYEAR .EQ. 0)GO TO 10                                     DAY
C                                                                  DAY
C                                                                  DAY
C                                                                  DAY
      IF (MONTH.NE.0) IDAY=IDAY+IMONTH(MONTH)                      DAY
C                                                                  DAY
    5 RETURN                                                       DAY
   10 IF(MONTH .NE. 0) IDAY = IDAY + IMON(MONTH)                   DAY
      GO TO 5                                                      DAY
C                                                                  DAY
      END                                                          DAY
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
• FLT DEGMOD,1,750925, 58801    , 1
      SUBROUTINE DEGMOD(RAD,IDEG)                                       DEGMOD
C                                                                       DEGMOD
C                                                                       DEGMOD   **=6
C     DEGMOD CONVERTS RADIANS INTO DEGREES, MINUTES, AND SECONDS        DEGMOD
C            AND MODS  THE DEGREES BETWEEN -360 TO +360                 DEGMOD
C                                                                       DEGMOD
C     IDEG(1)=DEGREES                                                   DEGMOD
C     IDEG(2)=MINUTES                                                   DEGMOD
C     IDEG(3)=SECONDS                                                   DEGMOD
C     RAD    = RADIANS                                                  DEGMOD
C     RADIAN = DEGREES PER RADIAN                                       DEGMOD
C     RDUM   = R*4 TEMPORARY STORAGE                                    DEGMOD
C                                                                       DEGMOD
      INTEGER IDEG(3)                                                   DEGMOD
C                                                                       DEGMOD
      COMMON /CONS/                                                     CONS
     1ECCE,CTS,CRUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,    CONS
     2TRES,WE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,     CONS
     3ICASE,IPPTS(10),NLINE,NPAGE,INP,LINMAX,ISWTR,ISWR,NVEH,NUDAY,     CONS
     4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDEBG(4)                           CONS
      REAL NAME,NAME1                                                   CONS
      LOGICAL LIDEBG                                                    CONS
C                                                                       DEGMOD
C     CONVERT RADIANS TO DEGREES, MINUTES, AND SECONDS                  DEGMOD
C                                                                       DEGMOD
      RDUM = RAD*RADIAN                                                 DEGMOD
      IDEG(1)=RDUM                                                      DEGMOD
      RDUM=ABS(RDUM-IDEG(1))*60.                                        DEGMOD
      IDEG(2)=RDUM                                                      DEGMOD
      IDEG(3)=(RDUM-IDEG(2))*60.+.5                                     DEGMOD
C                                                                       DEGMOD
      IF(IDEG(3).LT.60) GO TO 10                                        DEGMOD
      IDEG(3)=IDEG(3)-60                                                DEGMOD
      IDEG(2)=IDEG(2)+1                                                 DEGMOD
C                                                                       DEGMOD
   10 IF(IDEG(2).LT.60) GO TO 20                                        DEGMOD
      IDEG(2)=IDEG(2)-60                                                DEGMOD
C                                                                       DEGMOD
      ISIGN=1                                                           DEGMOD
      IF(IDEG(1).LT.0) ISIGN=-1                                         DEGMOD
      IDEG(1)=IDEG(1)+ISIGN                                             DEGMOD
C                                                                       DEGMOD
   20 CONTINUE                                                          DEGMOD
C                                                                       DEGMOD
      IDEG(1)=MOD(IDEG(1),360)                                          DEGMOD
      RETURN                                                            DEGMOD
      END                                                               DEGMOD
```

```
@ ELT FJECT,1,760331, 59923    , 1

      SUBROUTINE FJECT(NLINFS)                                         EJECT
C     RESTORE PAGE AND PRINT PAGE HEADER                               EJECT
      COMMON /CONS/                                                    CONS
     1ECCF,CIS,CMUV,RADIUS,HEADER(20),NAME(2),NAME1(2),THOPI,ALPHAM,   CONS
     2THLS,WE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,    CONS
     3ICASE,IPPTS(10),NLINE,NPAGE,INP,LINMAX,ISRTP,ISRR,NVEH,NODAY,    CONS
     4IV1TIM(6),IV2TIM(6),KO,TURBIT,L1DFBG(4)                          CONS
      REAL NAME,NAME1                                                  CONS
      LOGICAL L1DFBG                                                   CONS
      IF(NLINFS .EQ. 0) RETURN                                         EJECT
      NLINE = NLINES                                                   EJECT
      NPAGE = NPAGE + 1                                                EJECT
      WRITE(KO,9010)                                                   EJECT
      WRITE(KO,9020)(HEADER(I),I=1,18),ICASE,NPAGE                           *NEW
      RETURN                                                           EJECT **-1
 9010 FORMAT(1H1)                                                      EJECT
 9020 FORMAT(1X,18A4,2X,16HLPP SIMULATION   ,5HCASE ,I4,2X,4HPAGE,I4)        *NEW
      END                                                              EJECT **-1
```

```
* ECT ERRDET,1,760331, 59928   , 1
      SUBROUTINE ERRDET                                                 ERRDET
C     THIS ROUTINE CHECKS ALL ERRORS ON INPUT DATA CARDS                ERRDET
      COMMON /CONS/                                                     CONS
     1ECLE,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,    CONS
     2TMES,WE,ORBTT1(6),ORBTT2(6),PI,CJ2,CHU,PI2,PI32,SA(3),RADIAN,     CONS
     3ICASE,IPPIS(10),NLINE,NPAGE,INP,LINMAX,ISWTR,ISWR,NVEH,NODAY,     CONS
     4IV1TIM(6),IV2TIM(6),KO,TORBIT,LIDEBS(4)                           CONS
      REAL NAME,NAME1                                                   CONS
      LOGICAL LIDEBG                                                    CONS
      COMMON /FLAG/                                                     FLAG
     1IFIRST,IVEH,ISTART,TEND,IDEND                                     FLAG
      COMMON /SWAT/                                                     SWAT
     1DLONMN(100),DLONMX(100),TIME1(100,5),DLONG1(100,3),INLAT(2),      SWAT
     2ISLAT(2),ISOSTR,NLAT,LATNO(100),IALT(100),ICRDI(17)               SWAT
      DATA I549/549/,I65/65/,I101/101/,P10/10./,I1963/63/,              SET1
     1R6650/6650./,R7700/7700./,RSM/.00001/,RSMI/.15/,RP65/65./,        ERRDET
     2R360/360./          ,I1/1/,I17/17/,I31/31/,I12/12/                ERRDET
      DIMENSION XTO(2),LTO(2)                                           SET2
      IERC = 0                                                          ERRDET
      IF(NODAY .GE. 1 .AND. NODAY .LE.I549)GO TO 10                     ERRDET
      CALL PAGER(2)                                                     ERRDET
      WRITE(KO,900)                                                     ERRDET
      IERC = IERC + 1                                                   ERRDET
   10 IF(INLAT(1) .GE. 0  .AND. INLAT(1) .LE. INLAT(2))GO TO 15         ERRDET
      GO TO 30                                                          ERRDET
   15 IF(INLAT(2) .LE. I65)GO TO 20                                     ERRDET
      GO TO 30                                                          ERRDET
   20 IF(ISLAT(1) .GE. 0 .AND. ISLAT(1) .LE. ISLAT(2))GO TO 25          SET1
      GO TO 30                                                          ERRDET
   25 IF(ISLAT(2) .LE. I65)GO TO 40                                     SET1
   30 CALL PAGER(3)                                                     ERRDET
      WRITE(KO,905)                                                     ERRDET
      IERC = IERC + 1                                                   ERRDET
   40 ILC =INLAT(2) - INLAT(1) + 1                                      ERRDET
      NLAT= ILC+(IABS(ISLAT(2))- IABS(ISLAT(1))+1)                      ERRDET
      IF(NLAT.LT. I101)GO TO 45                                         ERRDET
      CALL PAGER(2)                                                     ERRDET
      WRITE(KO,910)                                                     ERRDET
      IERC = IERC + 1                                                   ERRDET
   45 DO 55 I=1,10,2                                                    ERRDET
      J = I+1                                                           ERRDET
      IF(IPPIS(I) .EQ. 0 .AND. IPPIS(J) .EQ. 0) GO TO 60                ERRDET
      IF(IPPIS(I) .GE. 1 .AND. IPPIS(I) .LE. IPPIS(J))GO TO 50          ERRDET
      GO TO 53                                                          ERRDET
   50 IF(IPPIS(J) .LE.I549)GO TO 55                                     ERRDET
   53 CALL PAGER(3)                                                     ERRDET
      WRITE(KO,915)I,J                                                  ERRDET
      IERC = IERC + 1                                                   ERRDET
   55 CONTINUE                                                          ERRDET
   60 IF(IVEH .GE. 1.AND. IVEH .LE. 2)GO TO 65                          ERRDET
      CALL PAGER(2)                                                     ERRDET
      WRITE(KO,920)                                                     ERRDET
      IERC = IERC + 1                                                   ERRDET
   65 DO 80 I=1,3                                                       ERRDET
      J = I+1                                                           ERRDET
      IF(ABS(SA(I)).GE. 0..AND.ABS(SA(I)).LE. P10)GO TO 70              ERRDET
      GO TO 75                                                          ERRDET
   70 IF( I .EQ. 3)GO TO 90                                             ERRDET
```

```
      IF(    SA(I) .LT. SA(J))GO TO 90                               ERRDET
   75 CALL PAGER(2)                                                  ERRDET
      WRITE(KO,925)I,J                                               ERRDET
      IERC = IERC + 1                                                ERRDET
   80 CONTINUE                                                       ERRDET
   90 IF(IV1TIM(1) .LE. 11963)GO TO 93                                       *NEW
      IF(IVEH .LT. 2)GO TO 95                                                *NEW
      IF(IV2TIM(1) .GT. 11963)GO TO 95                                       *NEW
   93 CALL PAGER(2)                                                          *NEW
      WRITE(KO,930)                                                  ERRDET  **-2
      IERC = IERC + 1                                                ERRDET
   95 IF(ORBIT1(1) .GE. R6650 .AND. ORBIT1(1) .LE. R7700)GO TO 100   ERRDET
      GO TO 105                                                      ERRDET
  100 IF(IVEH .LT. 2)GO TO 110                                               *NEW
      IF(ORBIT2(1) .GE. R6650 .AND. ORBIT2(1) .LE. R7700)GO TO 110           *NEW
  105 CALL PAGER(2)                                                  ERRDET  **-1
      WRITE(KO,935)                                                  ERRDET
      IERC = IERC + 1                                                ERRDET
  110 IF(ORBIT1(2) .GE. RSM .AND. ORBIT1(2) .LE. RSM1)GO TO 115      ERRDET
      GO TO 120                                                      ERRDET
  115 IF(IVEH .LT. 2)GO TO 125                                               *NEW
      IF(ORBIT2(2) .GE. RSM .AND. ORBIT2(2) .LE. RSM1)GO TO 125              *NEW
  120 CALL PAGER(2)                                                  ERRDET  **-1
      WRITE(KO,940)                                                  ERRDET
      IERC = IERC + 1                                                ERRDET
  125 IF(ORBIT1(4) .GE. 0 .AND. ORBIT1(4) .LE. R360)GO TO 130        ERRDET
      GO TO 135                                                      ERRDET
  130 IF(IVEH .LT. 2)GO TO 140                                               *NEW
      IF(ORBIT2(4) .GE. 0..AND. ORBIT2(4) .LE. R360)GO TO 140                *NEW
  135 CALL PAGER(2)                                                  ERRDET  **-1
      WRITE(KO,945)                                                  ERRDET
      IERC = IERC + 1                                                ERRDET
  140 IF(ORBIT1(5) .GE. 0 .AND. ORBIT1(5) .LE. R360)GO TO 145        ERRDET
      GO TO 150                                                      ERRDET
  145 IF(IVEH .LT. 2)GO TO 155                                               *NEW
      IF(ORBIT2(5) .GE. 0. .AND. ORBIT2(5) .LE. R360)GO TO 155               *NEW
  150 CALL PAGER(2)                                                  ERRDET  **-1
      WRITE(KO,950)                                                  ERRDET
      IERC = IERC + 1                                                ERRDET
  155 IF(ABS(ORBIT1(6)).GT. RP65)GO TO 157                                   *NEW
      IF(IVEH .LT. 2)GO TO 160                                               *NEW
      IF(ABS(ORBIT2(6)) .LE. RP65)GO TO 160                                  *NEW
  157 CALL PAGER(2)                                                          *NEW
      WRITE(KO,955)                                                  ERRDET  **-3
      IERC = IERC + 1                                                ERRDET
  160 CONTINUE                                                               *NEW
  175 IF(IV1TIM(3) .GE. 1 .AND. IV1TIM(3) .LE. I31)GO TO 180         SETI    *-10
      GO TO 185                                                      ERRDET
  180 IF(IVEH .LT. 2)GO TO 187                                               *NEW
      IF(IV2TIM(3) .GE. 1 .AND. IV2TIM(3) .LE. I31)GO TO 187                 *NEW
  185 CALL PAGER(2)                                                  ERRDET  **-1
      WRITE(KO,965)                                                  ERRDET
      IERC = IERC + 1                                                ERRDET
  187 IF(IV1TIM(2) .GE. 1 .AND. IV1TIM(2) .LE. I12)GO TO 190         ERRDET
      GO TO 195                                                      ERRDET
  190 IF(IVEH .LT. 2)GO TO 200                                               *NEW
      IF(IV2TIM(2) .GE. 1 .AND. IV2TIM(2) .LE. I12)GO TO 200                 *NEW
  195 CALL PAGER(2)                                                  ERRDET  **-1
      WRITE(KO,970)                                                  ERRDET
      IERC = IERC + 1                                                ERRDET
```

```
  200 IF(IERC .EQ. 0)GO TO 210                                          ERRDET
      CALL PAGEP(2)                                                     ERRDET
      WRITE(KO,980)                                                     ERRDET
      STOP                                                              ERRDET
  210 RETURN                                                            ERRDET
  900 FORMAT(1H0,54H*** NODAY IS OUT OF RANGE. IT IS NOT BETWEEN 1 AND 3ERRDET
     149.)                                                              ERRDET
  905 FORMAT(1H0,107H*** INLAT OR ISLAT IS OUT OF RANGE. THEY ARE NOT BEERRDET
     1TWEEN 0 AND + 65 OR THE FIRST VALUE OF EITHER ONE IS NOT/29HLESS TSFTI
     2HAN OR = TO THE SECOND )                                          SFTI
  910 FORMAT(1H0,80H*** THE TOTAL NUMBER OF LATITUDE POINTS SPECIFIED BYERRDET
     1 ISLAT AND INLAT EXCEEDS 100)                                     ERRDET
  915 FORMAT(1H0, 10H*** IPPIS(,I2,12H) AND IPPIS(,I2, 76H) ARE OUT OF RERRDET
     1ANGE. THEY ARE NOT BETWEEN 1 AND 549 OR THE VALUE OF THE FIRST/   ERRDET
     225HENTRY IN A PAIR IS NOT LESS THAN OR = TO THE SECOND)           ERRDET
  920 FORMAT(1H0,31H*** IVER IS NOT BETWEEN 1 AND 2)                    ERRDET
  925 FORMAT(1H0,62H*** EITHER AN SA ENTRY IS NOT BETWEEN 0 AND ABS(10) ERRDET
     1OR ENTRY (,I2,25H) IS NOT LESS THAN ENTRY(,I2,1H))                SFTI
  930 FORMAT(1H0,58H*** EITHER IV1TIM(1) OR IV2TIM(1) IS NOT GREATER THAERRDET
     1N 1963)                                                           ERRDET
  935 FORMAT(1H0,73H*** EITHER ORBIT1(1) OR ORBIT2(1) IS NOT BETWEEN 665ERRDET
     10 AND 7700 KILOMETERS)                                            ERRDET
  940 FORMAT(1H0,63H*** EITHER ORBIT1(2) OR ORBIT2(2) IS NOT BETWEEN .00ERRDET
     1001 AND .15)                                                      ERRDET
  945 FORMAT(1H0,58H*** EITHER ORBIT1(4) OR ORBIT2(4) IS NOT BETWEEN 0 AERRDET
     1ND 360)                                                           ERRDET
  950 FORMAT(1H0,58H*** EITHER ORBIT1(5) OR ORBIT2(5) IS NOT BETWEEN 0 AERRDET
     1ND 360)                                                           ERRDET
  955 FORMAT(1H0,59H*** EITHER ORBIT1(6) OR ORBIT2(6) IS NOT BETWEEN -65ERRDET
     1 AND 65)                                                          ERRDET
  965 FORMAT(1H0,50H*** IV1TIM(3) OR IV2TIM(3) IS NOT BETWEEN 1 AND 31) ERRDET   **-2
  970 FORMAT(1H0,50H*** IV1TIM(2) OR IV2TIM(2) IS NOT BETWEEN 1 AND 12) ERRDET
  980 FORMAT(1H0,78H*** THIS JOB IS ABANDONED DUE TO THE OCCURANCE OF 1 ERRDET
     1OR MORE FATAL INPUT ERRORS)                                       ERRDET
      END                                                               ERRDET
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
 @ ELT HECTOR,1,760331, 59920   , 1

      SUBROUTINE HECTOR                                                 HECTOR
C                                                                       HECTOR
C                                                                       HECTOR
C     SUBROUTINE HECTOR GENERATES THE EPHEMERIS DATA FOR TWO VEHICLES FOR HECTOR
C     ONE DAY.  THE EPHEMERIS IS MAINTAINED AS A SET OF ORBITAL ELEMENTS. HECTOR
C                                                                       HECTOR
C           INPUT --- SEMI-MAJOR AXIS (E.R.)                            HECTOR
C                 --- ECCENTRICITY                                      HECTOR
C                 --- INCLINATION (RAD)                                 HECTOR
C                 --- LONGITUDE OF THE NODE AT REF. TIME (RAD)          HECTOR
C                 --- ARGUMENT OF PERIGEE PASSAGE (RAD)                 HECTOR
C                 --- TIME OF PERIGEE PASSAGE (SEC)                     HECTOR
C                                OR                                     HECTOR
C                 --- LATITUDE + DIRECTION OF TRAVEL                    HECTOR
C                                                                       HECTOR
C                 --- TIME OF INJECTION (YR/MO/DY/HR/MN/SEC)            HECTOR
C                                                                       HECTOR
C           OUTPUT--- SEMI-MAJOR AXIS                                   HECTOR
C                 --- ECCENTRICITY                                      HECTOR
C                 --- INCLINATION                                       HECTOR
C                 --- LONGITUDE OF THE NODE                             HECTOR
C                 --- ARGUMENT OF PERIGEE                               HECTOR
C                 --- TIME OF PERIGEE PASSAGE                           HECTOR
C                 --- START TIME OF REV                                 HECTOR
C                 --- RIGHT ASC. OF NODE                                HECTOR
C                 --- PRECESSION RATE OF LINE OF APSIDES                HECTOR
C                 --- REGRESSION RATE OF NODE                           HECTOR
C                 --- ORBIT PLANE SUN ANGLE                             HECTOR
C                 --- ORBIT PLANE SUN ANGLE RATE                        HECTOR
      LOGICAL   ICHAR(4)                                                HECTOR
      DIMENSION IREVCT(17)                                              HECTOR
      DATA IREVCT/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17/            HECTOR
      INTEGER   IFLAG                                                   HECTOR
      DATA IFLAG/1/                                                     HECTOR
C                                                                       HECTOR
C                                                                       HECTOR
      COMMON/ICARUS/ CARUS(8,17,2)                                      HECTOR
      COMMON /ACCNUM/ ACCNUM(100)                                       SETI
      EQUIVALENCE (ACCNUM(1),KVEH)                                      SETI
      COMMON /CONS/                                                     CONS
     1ECCF,C1S,ENUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,    CONS
     2TOLS,KE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,     CONS
     3ICASE,IPPTS(10),NLINE,NPAGE,INP,LINMAX,ISWTH,ISWR,NVEH,NUDAY,     CONS
     4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDFBG(4)                           CONS
      REAL NAME,NAME1                                                   CONS
      LOGICAL LIDFBG                                                    CONS
C                                                                       HECTOR
      COMMON /HECCOM/                                                   HECCOM
     1ALPHAT,XBETAD(2),XBETA(2,2),XWD(2),XOMEGD(2),TASCNP(2),XTO(2),    HECCOM
     2PERIOD(2),LMO,LDA,NREV1,NREV2,IYR,IYRB,IDAY,IPDAY,ICDAY,NREV1P,   HECCOM
     3NREV2P,IV1DAY,IV2DAY,LYR                                          SETI
      COMMON /ACCEPH/                                                   ACCEPH
     1IPROOF(17,2),SM(1,2),EC(1,2),XIN(1,2),RTASC(1,2),WP(1,2),TNDE(1,2)ACCEPH
     2,PC(1)                                                            ACCEPH
      EQUIVALENCE (IO,ICHAR(1))                                         HECTOR
      DATA IO/0/                                                        HECTOR
C                                                                       HECTOR
      DIMENSION XTI(2)                                                  HECTOR
```

```
C                                                                        HECTOR
C  COMPUTE YEAR-1900,MONTH,DAY(FLT. PT.) FOR INPUT TO LFPA              HECTOR
C                                                                        HECTOR
C  TEST FOR END OF YEAR                                                  HECTOR
C                                                                        HECTOR
      IF(IDAY.LT.IPDAY) LYR=LYR+1                                        HECTOR
C                                                                        HECTOR
C  CALL CLDAY TO COMPUTE LMONTH AND LDAY                                 HECTOR
C                                                                        HECTOR
      CALL CLDAY                                                         HECTOR
C                                                                        HECTOR
C  UPDATE IPDAY FOR NEXT CALL TO HECTOR                                  HECTOR
C                                                                        HECTOR
      IF(IDAY .LT. IPDAY)IPDAY = IDAY - 1                                SET1
C                                                                        HECTOR
      FLDA=LDA                                                           HECTOR
      CALL LFPA(FLDA,LMO,LYR,ALFAGO,DAYS)                                HECTOR
      IVEH=KVEH                                                          HECTOR
      IREV = 1                                                           HECTOR
      IF(ICDAY .GT. 1) GO TO 202
      IF(KVEH .EQ. 1)GO TO 10                                            HECTOR
      IF(IV2DAY .EQ. IDAY)GO TO 100                                      HECTOR
      GO TO 202                                                          HECTOR
   10 IF(IV1DAY.EQ.IDAY) GO TO 100                                       HECTOR
      GO TO 202                                                          HECTOR
  100 CONTINUE                                                           HECTOR
      IF(KVEH .EQ. 1)GO TO 105                                           HECTOR
      XTO(IVEH)=IV2TIM(4)*3600+IV2TIM(5)*60+IV2TIM(6)                    HECTOR
      GO TO 40                                                           HECTOR
  105 XTO(IVEH)=IV1TIM(4)*3600+IV1TIM(5)*60+IV1TIM(6)                    HECTOR
  110 CONTINUE                                                           HECTOR
   40 CONTINUE                                                           HECTOR
C                                                                        HECTOR
C  KEPLERIAN ORBIT ELEMENTS                                              HECTOR
C                                                                        HECTOR
C           CARUS(1,1,IVEH)=SEMI-MAJOR AXIS (E.R.)                       HECTOR
C           CARUS(2,1,IVEH)=ECCENTRICITY (N.D.)                          HECTOR
C           CARUS(3,1,IVEH)=INCLINATION (RAD)                            HECTOR
C           CARUS(4,1,IVEH)=LONG. OF ASC. NODE (RAD)                     HECTOR
C           CARUS(5,1,IVEH)=ARG. OF PERIFOCUS (RAD)                      HECTOR
C           CARUS(6,1,IVEH)=TIME OF PERIFOCUS PASSAGE (SEC)              HECTOR
C           CARUS(7,1,IVEH)=TIME AT START OF THE REV (SEC)               HECTOR
C           CARUS(8,1,IVEH)=RIGHT ASCENSION OF THE NODE (RAD)            HECTOR
C           CARUS(9,1,IVEH) (NOT USED)                                   HECTOR
C           CARUS(10,1,IVEH, NOT USED)                                   HECTOR
C                                                                        HECTOR
C  COMPUTE ORBITAL ELEMENTS FOR FIRST REV                                HECTOR
C                                                                        HECTOR
C  MEAN MOTION                                                           HECTOR
C                                                                        HECTOR
      XN = SQRT(CMU/CARUS(1,1,IVEH)**3)                                  HECTOR
      P = TWOPI/XN                                                             *NEW
      KP = 1                                                                   *NEW
   50 XTEM      = XTO(IVEH) -KP*P                                              *NEW
      IF(XTEM       .LT. 0.)GO TO 60                                           *NEW
      KP = KP + 1                                                              *NEW
      GO TO 50                                                                 *NEW
   60 KP = KP - 1                                                              *NEW
      XTO(IVEH) = XTEM       +P                                                *NEW
      DELOMG = PI*P/43200.                                                     *NEW
```

```
      CARUS(4,1,IVEH) = CARUS(4,1,IVEH) + KP*DELOMG
      IF(CARUS(4,1,IVEH) .LT. TWOPI)GO TO 70                            *NEW
      CARUS(4,1,IVEH) = CARUS(4,1,IVEH) - DELOMG                        *NEW
      KP = KP + 1                                                       *NEW
   70 CONTINUE                                                          *NEW
      XECC = CARUS(2,1,IVEH)                                            *NEW
C  FORM (1.-ECC**2) + (1.-ECC**2)**1/2                           HECTOR
      TERM1 = 1. - XECC**2                                       HECTOR
      TERM2 = SQRT(TERM1)                                        HECTOR
C                                                                HECTOR
C  TIME OF PERIGEE PASSAGE INPUT. IF NOT,COMPUTE INITIAL VALUE FROM LATIHECTOR
C                                                                HECTOR
      IF(IORBIT.EQ.0) GO TO 150                                  HECTOR
C                                                                HECTOR
C  COMPUTE TIME OF PERIGEE PASSAGE FROM LATITUDE                 HECTOR
C                                                                HECTOR
      XLATI=CARUS(6,1,IVEH)                                      HECTOR
      XI=CARUS(3,1,IVEH)                                         HECTOR
      XWI = ASIN(SIN(XLATI)/SIN(XI))                             HECTOR
C                                                                HECTOR
C                                                                HECTOR
C  INPUT TYPE FLAG                                               HECTOR
C        FLAG            INPUT TYPE                              HECTOR
C          0       TIME OF PERIGEE PASSAGE (FOR BOTH VEHICLES)   HECTOR
C          1       LAT. OF INJECTION INPUT - ASC.  (BOTH VEH.)   HECTOR
C          2       LAT. OF INJECTION INPUT - DESC. (BOTH VEH.)   HECTOR
C          3       LAT. OF INJECTION INPUT - ASC.-VEH 1 + DES.-VEH 2 HECTOR
C          4       LAT. OF INJECTION INPUT - DES.-VEH 1 + ASC.-VEH 2 HECTOR
C                                                                HECTOR
C  ASCENDING                                                     HECTOR
C                                                                HECTOR
      IF (IORBIT .EQ. 1 .AND. XLATI .LT. 0.) XWI = TWOPI + XWI   HECTOR
      IF(IORBIT.EQ.3.AND.IVEH.EQ.1.AND.XLATI.LT.0.) XWI=TWOPI+XWI HECTOR
      IF(IORBIT.EQ.4.AND.IVEH.EQ.2.AND.XLATI.LT.0.) XWI=TWOPI+XWI HECTOR
C                                                                HECTOR
C  DESCENDING                                                    HECTOR
C                                                                HECTOR
      IF(IORBIT.EQ.2) XWI=PI-XWI                                 HECTOR
      IF(IORBIT.EQ.3.AND.IVEH.EQ.2) XWI = PI-XWI                 HECTOR
      IF(IORBIT.EQ.4.AND.IVEH.EQ.1) XWI = PI-XWI                 HECTOR
      XNUI=XWI-CARUS(5,1,IVEH)                                   HECTOR
C                                                                HECTOR
C  TEST FOR NU LESS THAN ZERO                                    HECTOR
C                                                                HECTOR
      IF(XNUI .LT. 0.)XNUI = TWOPI + XNUI                        HECTOR
      COSEI=(1./XECC)*(1.-TERM1/(1.+XECC*COS(XNUI)))                    *NEW
      XEI = ASIN(SIN(XNUI)*(1.-XECC*COSEI)/TERM2)                HECTOR **-1
      IF(COSEI .LT. 0.)XEI = PI - XEI                            HECTOR
      IF(XEI .LT. 0. .AND. COSEI .GT. 0.)XEI = TWOPI + XEI              *NEW
      DELTAU=(XEI-XECC*SIN(XEI))/XN                                     *NEW
      CARUS(6,1,IVEH)=XTO(IVEH)-DELTAU                           HECTOR
  150 CONTINUE                                                   HECTOR
C  SET PERIGEE PASSAGE TIME                                      HECTOR
C                                                                HECTOR
      XT = CARUS(6,1,IVEH)                                       HECTOR
C                                                                HECTOR
C                                                                HECTOR
C  MEAN ANOMALY                                                  HECTOR
C                                                                HECTOR
      XM=XN*(XTO(IVEH)-CARUS(6,1,IVEH))                          HECTOR
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C                                                                       HECTOR
C  CALL KEPLER TO COMPUTE ECC. ANOMALY                                  HECTOR
C                                                                       HECTOR
      CALL KEPLER(XM,XECC,XE,ERROR)                                     HECTOR
      IF(ERROR.EQ.0.) GO TO 120                                         HECTOR
      CALL PAGER(2)                                                     HECTOR
      WRITE(KO,1000)                                                    HECTOR
 1000 FORMAT(1H0,80H** ERROR (KEPLER) SOLUTION FOR ECC. ANOMALY DID NOT HECTOR
     1CONVERGE AFTER 50 ITERATIONS )                                    HECTOR
C                                                                       HECTOR
C                                                                       HECTOR
  120 CONTINUE                                                          HECTOR
      SINE = SIN(XE)                                                    HECTOR
      COSE = COS(XE)                                                    HECTOR
C                                                                       HECTOR
C  FORM (1.-ECC*COS(E))                                                 HECTOR
C                                                                       HECTOR
      TERM3 = 1. - XECC*COSE                                            HECTOR
C                                                                       HECTOR
C  TRUE ANOMALY (INJECTION)                                             HECTOR
C                                                                       HECTOR
      XNU = ASIN(TERM2*SINE/TERM3)                                      HECTOR
C                                                                       HECTOR
C  TEST FOR QUADRANT OF NU                                              HECTOR
C                                                                       HECTOR
      COSNU = (COSE-XECC)/(1.-XECC*COSE)                                HECTOR
C                                                                       HECTOR
      IF(COSNU .LT. 0.) XNU = PI - XNU                                  HECTOR
      IF(XNU .LT. 0. .AND. COSNU .GT. 0.) XNU = TWOPI + XNU             HECTOR
C                                                                       HECTOR
C  OTHERWISE NU IS UNCHANGED....                                        HECTOR
C                                                                       HECTOR
C  INPLANE ANGLE FROM ASC. NODE TO INJ. POINT                           HECTOR
C                                                                       HECTOR
      XU = PIMOD(CARDS(5,I,IVEH)+XNU)                                   HECTOR
C                                                                       HECTOR
C  INPLANE ANGLE FROM INJ. POINT TO START OF NEXT REV.(SOUTHERNMOST PT.)HECTOR
C                                                                       HECTOR
      DXNU = 3. * PI2 - XU                                              HECTOR
C                                                                       HECTOR
C  TRUE ANOMALY AT S. P.                                                HECTOR
C                                                                       HECTOR
      XNU1 = PIMOD(XNU + DXNU)                                          HECTOR
C                                                                       HECTOR
C  COSINE OF E AT NU                                                    HECTOR
C                                                                       HECTOR
      COSE1 = (1./XECC)*(1.-TERM1/(1.+XECC*COS(XNU1)))                  HECTOR
C                                                                       HECTOR
C  ECC. ANOMALY AT NU1                                                  HECTOR
C                                                                       HECTOR
      XE1 = ASIN(SIN(XNU1)*(1.-XECC*COSE1)/TERM2)                       HECTOR
      IF(COSE1 .LT. 0.) XE1 = PI - XE1                                  HECTOR
      IF(XE1 .LT. 0. .AND. COSE1 .GT. 0.) XE1 = TWOPI + XE1             HECTOR
C                                                                       HECTOR
C  NO. OF PERIGEE - PERIGEE REVS                                        HECTOR
C                                                                       HECTOR
      K = (XNU + DXNU)/TWOPI                                            HECTOR
      XK = K                                                            HECTOR
C                                                                       HECTOR
      TERM4 = TWOPI/XN                                                  HECTOR
```

```
      PERIOD(IVEH) = TERM4                                              HECTOR
C                                                                       HECTOR
C  TIME AT S. P. (T-S)                                                  HECTOR
C                                                                       HECTOR
      CARUS(7,1,IVEH)=(XE1-XECC*SIN(XE1))/XN+XT+XK*TERM4                HECTOR
C                                                                       HECTOR
C  TRUE ANOMALY AT ASCENDING NODE                                       HECTOR
C                                                                       HECTOR
      XNUNDE = XNU1 + PI2                                               HECTOR
C                                                                       HECTOR
C  ECCENTRIC ANOMALY AT ASCENDING NODE                                  HECTOR
C                                                                       HECTOR
      SINEND = SIN(XNUNDE)*(TERM1/(1.+XECC*COS(XNUNDE)))/TERM2          HECTOR
      XESTAR = ACOS((1.-TERM1/(1.+XECC*COS(XNUNDE)))/XECC)              HECTOR
      IF (SINEND .LT. 0.0) XESTAR = TWOPI - XESTAR                      HECTOR
C                                                                       HECTOR
C  TIME OF ASC. NODE FOR REV 1                                          HECTOR
C                                                                       HECTOR
      XT1(IVEH) = (XESTAR-XECC*SIN(XESTAR))/XN+XT+XK*TERM4              HECTOR
  140 IF( XT1(IVEH) .GE. CARUS(7,1,IVEH)) GO TO 145                     HECTOR
      XT1(IVEH) = XT1(IVEH) + TERM4                                     HECTOR
      GO TO 140                                                         HECTOR
C                                                                       HECTOR
  145 CONTINUE                                                          HECTOR
C                                                                       HECTOR
      TMNODE(1,IVEH) = XT1(IVEH)                                        HECTOR
C                                                                       HECTOR
C  RECESSION OF NODE + PREC. OF LINE OF APSIDES                         HECTOR
C                                                                       HECTOR
      TERM5 = (-3./2.)*(XN*CJ2)/((CARUS(1,1,IVEH)**2)*TERM1**2)         HECTOR
      XOMEGD(IVEH) = TERM5*COS(CARUS(3,1,IVEH))                         HECTOR
      XWD(IVEH)=TERM5*(5.*SIN(CARUS(3,1,IVEH))**2/2.-2.)                HECTOR
C                                                                       HECTOR
C  LONG. OF NODE + ARG. OF PERIGEE AT TIME OF NODE OF FIRST REV         HECTOR
C                                                                       HECTOR
      CARUS(4,1,IVEH)=PIMOD(CARUS(4,1,IVEH)+(XOMEGD(IVEH)-WE)*(XT1(IVEH)HECTOR
     1                 -XT0(IVEH)))                                     HECTOR
      CARUS(5,1,IVEH)=PIMOD(CARUS(5,1,IVEH)+XWD(IVEH)*(XT1(IVEH)-XT0(IVEHECTOR
     *H)))                                                              HECTOR
      IF(CARUS(5,1,IVEH).GT.PI32) CARUS(5,1,IVEH) = CARUS(5,1,IVEH)     HECTOR
     *                                                -TWOPI            HECTOR
C                                                                       HECTOR
C  TIME OF PERIGEE PASSAGE FOR FIRST REV                                HECTOR
C                                                                       HECTOR
      CARUS(6,1,IVEH)=CARUS(6,1,IVEH)+(XK+1.)*TERM4                     HECTOR
C                                                                       HECTOR
C  RIGHT ASCENSION OF NODE AT TIME OF NODAL CROSSING                    HECTOR
C                                                                       HECTOR
      CARUS(8,1,IVEH)=PIMOD(ALFAG0+WE*XT1(IVEH)+CARUS(4,1,IVEH))        HECTOR
C                                                                       HECTOR
C  CALL SUBROUTINE ALPHA TO COMPUTE RIGHT ASC. OF SUN AT TIME OF 1ST REVHECTOR
C                                                                       HECTOR
      CALL ALPHA(IFLAG)                                                 HECTOR
C                                                                       HECTOR
C     -ORBIT PLANE SUN ANGLE -- BETA                                    HECTOR
C                                                                       HECTOR
      XBETA(1,IVEH)=PIMOD(ALPHAT-CARUS(8,1,IVEH))                       HECTOR
C                                                                       HECTOR
C     -ORBIT PLANE SUN ANGLE -- BETA          0 HOUR                    HECTOR
C                                                                       HECTOR
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      XOMGPR = CARUS(4,1,IVEH)+(XOMEGD(IVEH)-WE)*(-XT1(IVEH))          HECTOR
      XBETA(2,IVEH) = PI-XOMGPR                                        HECTOR
      IF(XOMGPR.GT.PI) XBETA(2,IVEH) = 3.*PI-XOMGPR                    HECTOR
C                                                                      HECTOR
C     -RATE OF CHANGE OF ANGLE -- BETA-DOT                             HECTOR
C                                                                      HECTOR
      XBETAD(IVEH) = (XBETA(1,IVEH)-XBETA(2,IVEH))/XT1(IVEH)           HECTOR
C                                                                      HECTOR
C                                                                      HECTOR
      GO TO 211                                                        HECTOR
C                                                                      HECTOR
C                                                                      HECTOR
C  SET ELEMENT                                                         HECTOR
C                                                                      HECTOR
  202 CONTINUE                                                         HECTOR
C                                                                      HECTOR
C  SET REV COUNT FROM PREVIOUS DAY                                     HECTOR
C                                                                      HECTOR
      NREVP = NREV1P                                                   HECTOR
      IF(IVEH.EQ.2) NREVP=NREV2P                                       HECTOR
      CARUS(1,1,IVEH) = CARUS(1,NREVP,IVEH)                            HECTOR
      XN = SQRT(CMU/CARUS(1,1,IVEH)**3)                                HECTOR
      CARUS(2,1,IVEH) = CARUS(2,NREVP,IVEH)                            HECTOR
      CARUS(3,1,IVEH) = CARUS(3,NREVP,IVEH)                            HECTOR
      TERM4 = PERIOD(IVEH)                                             HECTOR
      CARUS(4,1,IVEH) = PIMOD(CARUS(4,NREVP,IVEH)+(XOMEGD(IVEH)-WE)    HECTOR
     1                  *TERM4)                                        HECTOR
      CARUS(5,1,IVEH) = PIMOD(CARUS(5,NREVP,IVEH)+XWD(IVEH)            HECTOR
     1                  *TERM4)                                        HECTOR
      IF(CARUS(5,1,IVEH).GT.PI32) CARUS(5,1,IVEH) = CARUS(5,1,IVEH)    HECTOR
     *                                             -TWOPI              HECTOR
C                                                                      HECTOR
C  TEST FOR SECOND DAY OF LAUNCH FOR EITHER VEHICLE                    HECTOR
C                                                                      HECTOR
      IF(ICDAY .GT. 2) GO TO 206
      IF(IVEH.EQ.2) GO TO 204                                          HECTOR
      IF((IDAY-IV1DAY)-1) 206,205,206                                  HECTOR
  204 IF((IDAY-IV2DAY)-1) 206,205,206                                  HECTOR
C  COMPUTE TIME OF PERIGEE PASSAGE FOR DAY 2                           HECTOR
  205 CARUS(6,1,IVEH) = CARUS(6,NREVP,IVEH)+(TWOPI+(XWD(IVEH)*(86400.  HECTOR
     1                  -XT0(IVEH))))/XN-86400.                        HECTOR
      GO TO 208                                                        HECTOR
  206 CONTINUE                                                         HECTOR
C  COMPUTE TIME OF PERIGEE PASSAGE FOR DAY GREATER THAN DAY 2 OF MISSIONHECTOR
      CARUS(6,1,IVEH) = CARUS(6,NREVP,IVEH)+(TWOPI+ XWD(IVEH)*86400.)/XNHECTOR
     1                  -86400.                                        HECTOR
  208 CONTINUE                                                         HECTOR
      CARUS(7,1,IVEH) = CARUS(7,NREVP,IVEH)+TERM4-86400.               HECTOR
C                                                                      HECTOR
C  COMPUTE TIME OF ASC. NODE FOR FIRST REV OF DAY OTHER THAN DAY 1     HECTOR
C                                                                      HECTOR
      XT1(IVEH) = (TASCNP(IVEH)+NREVP*TERM4)-86400.                    HECTOR
C                                                                      HECTOR
      TMNODE(1,IVEH) = XT1(IVEH)                                       HECTOR
C                                                                      HECTOR
C  COMPUTE RT. ASC. OF NODE AT T1 FOR DAY OTHER THAN DAY 1             HECTOR
      CARUS(8,1,IVEH) = PIMOD(ALFAGO+WE*XT1(IVEH)+CARUS(4,1,IVEH))     HECTOR
  210 CONTINUE                                                         HECTOR
C                                                                      HECTOR
C  CALCULATE BETA-DOT FOR DAY OTHER THAN LAUNCH                        HECTOR
```

```
C
C  BETA FOR DAY - 1
C
      XBETA(2,IVEH)=XBETA(1,IVEH)
      CALL ALPHA(IFLAG)
C
C  BETA FOR THIS DAY
C
      XBETA(1,IVEH)=PIMOD(ALPHAT-CARUS(8,1,IVEH))
C
C  BETA DOT
C
      XBETAD(IVEH)=(XBETA(1,IVEH)-XBETA(2,IVEH))/(XTI(IVEH)
     1             +(86400.-TASCNP(IVEH)))
C
C  COMPUTE ORBIT ELEMENT FOR NREV .GT. 1
C
  211 DO 220 IREV=2,17
      IREVM1 = IREV-1
      TMNODE(IREV,IVEH) = TMNODE(IREVM1,IVEH)+TERM4
      CARUS(1,IREV,IVEH) = CARUS(1,IREVM1,IVEH)
      CARUS(2,IREV,IVEH) = CARUS(2,IREVM1,IVEH)
      CARUS(3,IREV,IVEH) = CARUS(3,IREVM1,IVEH)
      CARUS(4,IREV,IVEH) = PIMOD(CARUS(4,IREVM1,IVEH)+(XOMEGD(IVEH)-WE)
     1                     *TERM4)
      CARUS(5,IREV,IVEH) = PIMOD(CARUS(5,IREVM1,IVEH)+XWD(IVEH)
     1                     *TERM4)
      IF(CARUS(5,IREV,IVEH).GT.PI52) CARUS(5,IREV,IVEH)
     * = CARUS(5,IREV,IVEH)-TWOPI
      CARUS(6,IREV,IVEH) = CARUS(6,IREVM1,IVEH)+TERM4
      CARUS(7,IREV,IVEH) = CARUS(7,IREVM1,IVEH)+TERM4
      CARUS(8,IREV,IVEH) = PIMOD(CARUS(8,IREVM1,IVEH)+XOMEGD(IVEH)
     1                     *TERM4)
      IF (TMNODE(IREV,IVEH).GT.86400.) GO TO 300
  220 CONTINUE
      GO TO 310
  300 IREV = IREVM1
  310 CONTINUE
C
C  SET REV COUNT
C
      IF(IVEH.EQ.1) NREV1=IREV
      IF(IVEH.EQ.2) NREV2=IREV
C
C  SET TIME OF ASC. NODE (FIRST) OF THIS DAY FOR NEXT CALL TO HECTOR
C
C
      TASCNP(IVEH) = XTI(IVEH)
  400 CONTINUE
C
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE INPROC                                                 INPROC
C     THIS ROUTINE READS THE CARD INPUT,STORES THE DATA,AND PERFORMS INIINPROC
C     TIAL CALCULATIONS.                                                INPROC
      COMMON /CONS/                                                     CONS
     1ECCE,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,    CONS
     2THES,FE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,     CONS
     3ICASE,IPPI5(10),NLINE,NPAGE,INP,LINMAX,ISNTR,ISNR,NVEH,NUDAY,     CONS
     4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDFBG(4)                           CONS
      REAL NAME,NAME1                                                   CONS
      LOGICAL LIDFBG                                                    CONS
      COMMON /FLAG/                                                     FLAG
     1IFIRST,IVEH,ISTART,IEND,IDEND                                     FLAG
      COMMON /SWAT/                                                     SWAT
     1DLONMN(100),DLONMX(100),TIME1(100,3),DLONG1(100,3),INLAT(2),      SWAT
     2ISLAT(2),ISOSTR,NLAT,LATNO(100),IALT(100),IGRDI(17)               SWAT
      DIMENSION INLATT(2),ISLATT(2),IPPI5T(10)                          INPROC
      READ(INP,900)(HEADER(I),I=1,20)                                   INPROC
      READ(INP,905)ICASE,IVEHT,NUDAYT,(INLATT(I),I=1,2),(ISLATT(I),I=1,2INPROC
     1),(LIDFBG(I),I=1,4),(IPPI5T(I),I=1,10),ISOSTR,IDUM                INPROC
      READ(INP,910)(SA(I),I=1,3),(IV1TIM(I),I=1,6),(IV2TIM(I),I=1,6),   INPROC   **-1
     1IDUM                                                              INPROC
      READ(INP,915)(ORBIT1(I),I=1,6),(ORBIT2(I),I=1,6),IDUM             INPROC
      IF(NUDAYT .NE. 0)NUDAY = NUDAYT                                   INPROC
      IF(IVEHT .NE. 0)IVEH = IVEHT                                      INPROC
      IF(INLATT(1) .NE. 0)INLAT(1) = INLATT(1)                          INPROC
      IF(INLATT(2) .NE. 0)INLAT(2) = INLATT(2)                          INPROC
      IF(ISLATT(1) .NE. 0)ISLAT(1) = ISLATT(1)                          INPROC
      IF(ISLATT(2) .NE. 0)ISLAT(2) = ISLATT(2)                          INPROC
      DO 5 I=1,10                                                       INPROC
      IF(IPPI5T(I) .NE. 0)IPPI5(I) = IPPI5T(I)                          INPROC
    5 CONTINUE                                                          INPROC
      WRITE(KO,930)(HEADER(I),I=1,18)                                            *NEW
      WRITE(KO,935)ICASE,IVEH,NUDAY,INLAT,ISLAT,LIDFBG,IPPI5,ISOSTR              *NEW
      WRITE(KO,940)SA,IV1TIM,IV2TIM                                              *NEW
      WRITE(KO,945)ORBIT1(1),ORBIT1(2),(ORBIT1(I),I=4,6),ORBIT2(1),              *NEW
     1ORBIT2(2),(ORBIT2(I),I=4,6)                                                *NEW
      CALL EJECT(2)                                                              *NEW
      CALL ERROFI                                                       INPROC
      ORBIT1(1)  = ORBIT1(1)/RADIUS                                     INPROC
      ORBIT2(1)  = ORBIT2(1)/RADIUS                                     INPROC
      DO 10 I=3,6                                                       INPROC
      ORBIT1(I) = ORBIT1(I)/RADIAN                                      INPROC
      ORBIT2(I) = ORBIT2(I)/RADIAN                                      INPROC
   10 CONTINUE                                                          INPROC
      DO 15 I=1,3                                                       INPROC
      SA(I) = SA(I)/RADIAN                                              INPROC
   15 CONTINUE                                                          INPROC
      NVEH = IVEH                                                       INPROC
      IORBIT = 2                                                        INPROC
      TEMP = (-4.*PI*(ORBIT1(1))**2)*(1. - ORBIT1(2)**2)**2             INPROC
      TEMP1 = 3.* SQRT(CMU/(ORBIT1(1))**3)*365.25*86400.*CJ2            INPROC
      TEMP2 = TEMP/TEMP1                                                SET1
      ORBIT1(3) = ATAN(SQRT(1.-TEMP2**2)/TEMP2) + PI                    SET1
      ORBIT1(4) = ORBIT1(4) + PI -ASIN(-TAN(ORBIT1(6))/TAN(ORBIT1(3)))           *NEW
      IF(ORBIT1(4) .GT. TWOPI)ORBIT1(4) = ORBIT1(4) - TWOPI                      *NEW
      XT01 = IV1TIM(4)*3600 + IV1TIM(5)*60 + IV1TIM(6)                           *NEW
      LT01 = XT01 - 43200. + ORBIT1(4)*43200./PI                                 *NEW
      IF(LT01 .GE. 25200 .AND. LT01 .LE. 61200)GO TO 100                         *NEW
```

```
   95 CALL PAGER(2)                                                          *NEW
      WRITE(KO,960)                                                          *NEW
      STOP                                                                   *NEW
  100 IF(IVEH .LT. 2)GO TO 110                                               *NEW
      TEMP = (-4.*PI*(ORBIT2(1))**2)*(1. - ORBIT2(2)**2)**2          INPROC
      TEMP1 = 3.* SQRT(CMU/(ORBIT2(1))**3)*365.25*86400.*CJ2         INPROC
      TEMP2 = TEMP/TEMP1                                             SFTT
      ORBIT2(3) = ATAN(SQRT(1.-TEMP2**2)/TEMP2) + PI                 SFTT
      ORBIT2(4) = ORBIT2(4) + PI -ASIN(-TAN(ORBIT2(6))/TAN(ORBIT2(3)))       *NEW
      IF(ORBIT2(4) .GT. TWOPI)ORBIT2(4) = ORBIT2(4) - TWOPI                  *NEW
      XT02 = IV2TIM(4)*3600 + IV2TIM(5)*60 + IV2TIM(6)                       *NEW
      LT02 = XT02 - 43200. + ORBIT2(4)*43200/PI                              *NEW
      IF(LT02 .GE. 25200 .AND. LT02 .LE. 61200)GO TO 110                     *NEW
      GO TO 95                                                               *NEW
  110 LATNO(1) = INLAT(2)                                                    *NEW
      DO 600 I=2,66                                                  INPROC  **-1
      IK = I                                                         INPROC
      ITEM      = LATNO(I-1) - 1                                     INPROC
      IF(ITEM .LE. INLAT(1))GO TO 610                                INPROC
      LATNO(I) = ITEM                                                INPROC
  600 CONTINUE                                                       INPROC
  610 LATNO(IK) = INLAT(1)                                           INPROC
      IK = IK + 1                                                    INPROC
      LATNO(IK) = -ISLAT(1)                                          SFTT
      DO 620 I=1,65                                                  INPROC
      ITEM  = LATNO(I+IK-1) - 1                                      INPROC
      IK1 = I                                                        INPROC
      IF(ITEM .LE. - ISLAT(2))GO TO 630                              SFTT
      LATNO(I+IK) = ITEM                                             INPROC
  620 CONTINUE                                                       INPROC
  630 LATNO(IK+IK1) = -ISLAT(2)                                      SFTT
      RETURN                                                         INPROC
  900 FORMAT(20A4)                                                   INPROC
  905 FORMAT(I4,I2,I3,4I2,4I1,10I3,I2,25X,I2)                        INPROC
  910 FORMAT(3F5.1,12I2,32X,I2)                                              *NEW
  915 FORMAT(2(F6.1,F6.5,4F4.0),22X,I2)                              INPROC  **-1
  930 FORMAT(1H1,56X,6HHEADER/1H ,5X,18A4)                                   *NEW
  935 FORMAT(1H0,5X,95HICASE  IVEH  NODAY  INLAT   ISLAT   LIDEBG            *NEW
     1     IPRTS START/STOP TIMES                  ISOSTR/                   *NEW
     17X,I4,3X,I2,4X,I3,5X,                                                  *NEW
     24(I2,2X),1X,4I1,3X,5(I3,1X,I3,2X),2X,I2,5H   01)                       *NEW
  940 FORMAT(1H0,13X,45HSA                    IV1TIM             IV2TIM/6X   *NEW
     1,3(F5.1,2X),1X,6(I2,1X),2X,6(I2,1X),36X,2H02)                          *NEW
  945 FORMAT(1H0,21X, 6HORBIT1,30X,6HORBIT2/6X,  F6.1,2X,F6.5,2X,            *NEW
     13(F4.0,2X),1X,                                                         *NEW
     1 ,F6.1,2X,F6.5,2X,3(F4.0,2X),27X,2H03)                                 *NEW
  960 FORMAT(1H0,96H*** IV1TIM(4) OR IV2TIM(4) SPECIFIES A LOCAL VEHICLE     *NEW
     1 PASSAGE TIME NOT BETWEEN 700 AND 1700 HOURS)                          *NEW
      END                                                            INPROC
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
* FLT 10,1,750925, 58805 ,11

000001          UHUF
000002          DR 8,BL=1500
000003          DR 9,BL=1000
```

KEPLER,1,750925, 58787 , 1

```
      SUBROUTINE KEPLER(XM,XECC,XE,ERROR)                              KEPLER
C                                                                      KEPLER
C                                                                      KEPLER  **-B
C  THIS SUBROUTINE SOLVES KEPLER*S EQUATIONS FOR ECCENTRIC ANOMALY (E) KEPLER
C  GIVEN ECCENTRICITY (ECC) AND MEAN ANOMALY (M).                      KEPLER
C  THE EQUATIONS FOR THE ELLIPTIC CASE IS ---- M=E-ECC*SIN(E).         KEPLER
C                                                                      KEPLER
C  LOGIC FOR THE SOLUTION OF THE HYPERBOLIC CASE HAS BEEN REMOVED.     KEPLER
C                                                                      KEPLER
C  INPUT                                                               KEPLER
C          XM ---- MEAN ANOMALY (RAD)                                  KEPLER
C          XECC -- ORBIT ECCENTRICITY (N.D.)                           KEPLER
C  OUTPUT                                                              KEPLER
C          XE ---- ECCENTRIC ANOMALY (RAD)                             KEPLER
C          ERROR =1 ERROR FLAG INDICATING CONVERGENCE CONDITIONS.      KEPLER
C                =0 IMPLIES CONVERGENCE HAS OCCURRED.                  KEPLER
C                                                                      KEPLER
      DIMENSION B(4),C(4)                                              KEPLER
      COMMON /CONS/                                                    CONS
     1ECCF,C15,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,   CONS
     2THES,VE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,    CONS
     3ICASE,IPRIS(10),NLINE,NPAGE,INP,LINMAX,ISWTH,ISWR,NVER,NUDAY,    CONS
     4IV1TIM(6),IV2TIM(6),KN,IORBIT,LIDFBG(4)                          CONS
      REAL NAME,NAME1                                                  CONS
      LOGICAL LIDFBG                                                   KEPLER
      REAL     M                                                       KEPLER
      M = XM                                                           KEPLER
      ECC = XECC                                                       KEPLER
      ERROR = 0.                                                       KEPLER
      IF(M.NE.0.0) GO TO 9                                             KEPLER
      E = 0.                                                           KEPLER
      GO TO 8                                                          KEPLER
    9 CONTINUE                                                         KEPLER
C                                                                      KEPLER
C     CALCULATE INITIAL VALUE OF ECCENTRIC ANOMALY                     KEPLER
C                                                                      KEPLER
      TEMP = M                                                         KEPLER
      M = PIMOD(TEMP)                                                  KEPLER
C                                                                      KEPLER
C     FOR B(I) .LT. M .LE. B(I+1), I=0,1,2,3, THEN E0 = C(I+1)         KEPLER
C     IF M .GT. B(4), THEN E0 = (M-2*PI*ECC)/(1-ECC)       PI=3.14159..KEPLER
C        WHERE      B(0) IS MINUS INFINITY                             KEPLER
C                   B(1) = 1-ECC            C(1) = M/B(1)              KEPLER
C                   B(2) = PI-1-ECC         C(2) = M+ECC               KEPLER
C                   B(3) = PI+1+ECC         C(3) = (M+ECC*PI)/(1+ECC)  KEPLER
C                   B(4) = 2*PI-1+ECC       C(4) = (M-ECC)             KEPLER
      B(1) = 1.-ECC                                                    KEPLER
      B(2) = PI-1.-ECC                                                 KEPLER
      B(3) = PI+1.+ECC                                                 KEPLER
      B(4) = 2.*PI-1.+ECC                                              KEPLER
      C(1) = M/B(1)                                                    KEPLER
      C(2) = M+ECC                                                     KEPLER
      C(3) = (M+PI*ECC)/(1.+ECC)                                       KEPLER
      C(4) = M-ECC                                                     KEPLER
      DO 2 J=1,4                                                       KEPLER
      IF(M.GT.B(J)) GO TO 2                                            KEPLER
      OLDE=C(J)                                                        KEPLER
      GO TO 3                                                          KEPLER
```

```
    2 CONTINUE                                                          KEPLER
      OLDE=(M-2.0*PI*ECC)/(1.-ECC)                                      KEPLER
C                                                                       KEPLER
C     CALCULATE ECCENTRIC ANOMALY                                       KEPLER
C                                                                       KEPLER
    3 CONTINUE                                                          KEPLER
      DO 10 N=1,50                                                      KEPLER
      FOFE=SIN(OLDE)                                                    KEPLER
      DFOFF=COS(OLDE)                                                   KEPLER
      E=OLDE+(M-OLDE+ECC*FOFE)/(1.-ECC*DFOFF)                           KEPLER
      IF(ABS((OLDE-E)/OLDE).LE.1.0E-07) GO TO 7                         KEPLER
      OLDE=E                                                            KEPLER
   10 CONTINUE                                                          KEPLER
      ERROR=1.0                                                         KEPLER
    7 TEMP = E                                                          KEPLER
      E = PIMOD(TEMP)                                                   KEPLER
    8 XE=E                                                              KEPLER
      RETURN                                                            KEPLER
      END                                                               KEPLER
```

```
@ FLT LATSEL,1,750922, 69845    , 1

      SUBROUTINE LATSEL                                                 LATSEL
C     THIS ROUTINE RESTORES SWATH DATA IN I/O AREA AND COMPUTES MIN-MAX LATSEL
C     DELTA LONGITUDE TABLES                                            LATSEL
      COMMON /SWAT/                                                     SWAT
     1DLONMN(100),DLONMX(100),TIME1(100,3),DLONG1(100,3),INLAT(2),      SWAT
     2ISLAT(2),ISOSTR,NLAT,LATNO(100),IALT(100),ICRDI(17)               SWAT
      COMMON /REVSTH/                                                   REVSTH
     1TIME(181,3,2),DLONG(181,3,2),ALT(181,2)                           REVSTH
      COMMON /CONS/                                                     CONS
     1ECLF,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,    CONS
     2THES,WE,ORBIT1(6),ORBIT2(6),PI,CJ2,CHU,PI2,PI32,SA(5),RADIAN,     CONS
     3ICASE,IPPIS(10),NLINE,NPAGE,INP,LINMAX,ISWTH,ISWR,NVEH,NODAY,     CONS
     4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDFBG(4)                           CONS
      REAL NAME,NAME1                                                   CONS
      LOGICAL LIDFBG                                                    CONS
      COMMON /ACCNUM/ ACCNUM(100)                                       LATSEL
      EQUIVALENCE(ACCNUM(1),KVEH)                                       LATSEL
      DO 50 I=1,NLAT                                                    LATSEL
      IF(LATNO(I) .GE. 0)GO TO 5                                        LATSEL
      ISUB = IABS(LATNO(I))+91                                          LATSEL
      GO TO 10                                                          LATSEL
    5 ISUB =91 - LATNO(I)                                               SET2
   10 IALT(I) = (ALT(ISUB,KVEH))*10.*RADIUS                             SET2
      DO 15 K=1,3                                                       LATSEL
      TIME1(I,K) = TIME(ISUB,K,KVEH)                                    LATSEL
      DLONG1(I,K) = PIMOD(DLONG(ISUB,K,KVEH))                           SET2
   15 CONTINUE                                                          LATSEL
      OMEG1 = TWOPI + DLONG1(I,1) - 2*DLONG1(I,2)                       LATSEL
      OMEG2 = TWOPI + DLONG1(I,3) - 2*DLONG1(I,2)                       LATSEL
      IF(OMEG1 .GE. OMEG2)GO TO 20                                      LATSEL
      IF(OMEG2 .GT. DLONMX(I))DLONMX(I) = OMEG2                         LATSEL
      IF(OMEG1 .LT. DLONMN(I))DLONMN(I) = OMEG1                         LATSEL
      GO TO 50                                                          LATSEL
   20 IF(OMEG1 .GT. DLONMX(I))DLONMX(I) = OMEG1                         LATSEL
      IF(OMEG2 .LT. DLONMN(I))DLONMN(I) = OMEG2                         LATSEL
   50 CONTINUE                                                          LATSEL
      RETURN                                                            LATSEL
      END                                                               LATSEL
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
# FOR LFPA,1,750925, 58785 , 1

      SUBROUTINE LFPA(FLDA,LMO,LYR,ALFGM,DAYS)                        LFPA
C                                                                     LFPA
C                                                                     LFPA    **-7
C     GIVEN  DAY,MONTH,YEAR - SUBR. LFPA RETURNS THE RIGHT ASCENSION OF LFPA
C     GREENWICH AT MIDNIGHT OF A GIVEN DAY                            LFPA
C                                                                     LFPA
      COMMON /CONS/                                                   CONS
     1ECLF,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,  CONS
     2TIES,WE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,   CONS
     3ICASE,IPPIS(10),NLINE,NPAGE,INP,LINMAX,ISHIP,ISWR,NVER,NUDAY,   CONS
     4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDEBG(4)                         CONS
      REAL NAME,NAME1                                                 CONS
      LOGICAL LIDEBG                                                  CONS
C     REFERENCE EPOCH IS 0 HOUR 1JAN1950                              LFPA
C                                                                     LFPA
C     COMPUTE DAYS IN FULL YEARS FROM EPOCH TO LYR                    LFPA
C     DAYS TO 1JAN1963 IS 4748.                                       LFPA
C                                                                     LFPA
      DAYS=4748.                                                      LFPA
C     NOTE--LYR MUST BE GREATER THAN OR EQUAL TO (19)64               LFPA
      LASTYR=LYR-1                                                    LFPA
      DO 20 I=63,LASTYR                                               LFPA
      KREMD=MOD((I-60),4)                                             LFPA
      IF(KREMD.GT.0) GO TO 10                                         LFPA
      DAYS=DAYS+366                                                   LFPA
      GO TO 20                                                        LFPA
   10 DAYS=DAYS+365                                                   LFPA
   20 CONTINUE                                                        LFPA
      IF(LMO-1)30,40,30                                               LFPA
   30 IF(LMO-2)60,50,60                                               LFPA
   40 DAYS=DAYS+FLDA-1.                                               LFPA
      GO TO 270                                                       LFPA
   50 DAYS=DAYS+FLDA+30.                                              LFPA
      GO TO 270                                                       LFPA
   60 KDEL=MOD((LYR-60),4)                                            LFPA
      IF(KDEL.GT.0) GO TO 70                                          LFPA
      DAYS=DAYS+59.                                                   LFPA
      GO TO 80                                                        LFPA
   70 DAYS=DAYS+58.                                                   LFPA
   80 IF(LMO-3) 270,170,90                                            LFPA
   90 IF(LMO-5) 180,190,100                                           LFPA
  100 IF(LMO-7) 200,210,110                                           LFPA
  110 IF(LMO-9) 220,230,120                                           LFPA
  120 IF(LMO-11)240,250,260                                           LFPA
  170 DAYS=DAYS+FLDA                                                  LFPA
      GO TO 270                                                       LFPA
  180 DAYS=DAYS+FLDA+31.                                              LFPA
      GO TO 270                                                       LFPA
  190 DAYS=DAYS+FLDA+61.                                              LFPA
      GO TO 270                                                       LFPA
  200 DAYS=DAYS+FLDA+92.                                              LFPA
      GO TO 270                                                       LFPA
  210 DAYS=DAYS+FLDA+122.                                             LFPA
      GO TO 270                                                       LFPA
  220 DAYS=DAYS+FLDA+153.                                             LFPA
      GO TO 270                                                       LFPA
  230 DAYS=DAYS+FLDA+184.                                             LFPA
      GO TO 270                                                       LFPA
```

```
249 DAYS=DAYS+FLDA+214.                                           LFPA
    GO TO 270                                                     LFPA
250 DAYS=DAYS+FLDA+245.                                           LFPA
    GO TO 270                                                     LFPA
260 DAYS=DAYS+FLDA+275.                                           LFPA
270 CONTINUE                                                      LFPA
    ALEGM=AMOD(SNGL((.10007554203+(.985647346D0)*DAYS+            LFPA
   1(2.9015D-13)*DAYS**2))+360.)                                  LFPA
    ALEGO=ALEGM/RADIAN                                            LFPA
    RETURN                                                        LFPA
    END                                                           LFPA
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
 • FLT MAIN,1,760331, 59925       , 1

C     PROGRAM MAIN(ITEST=401,OTEST=401,TAPE5=ITEST,TAPE6=OTEST,      SET7
C    1SWATH=1500,SWATHR=500,TAPE8=SWATH,TAPE9=SWATHR)                SET7
C     THIS IS THE MAIN CONTROL ROUTINE FOR EPHEM PROGRAM             MAIN
      COMMON /FLAG/                                                   FLAG
     1IFIRST,IVEH,ISTART,IEND,IDEND                                   FLAG
      COMMON /CONS/                                                   CONS
     1ECCE,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),THOPT,ALPHAM,  CONS
     2THES,WE,ORBIT1(6),ORBIT2(6),PI,CJ2,CHU,PI2,PI32,SA(3),RADIAN,   CONS
     3ILATC,IPRIS(10),NLINE,NPAGE,INP,LINMAX,ISWTH,ISWR,NVEH,NODAY,   CONS
     4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDEBG(4)                         CONS
      REAL NAME,NAME1                                                 CONS
      LOGICAL LIDEBG                                                  CONS
      COMMON /HECCOM/                                                 HECCOM
     1ALPHAT,XBETAO(2),XBETA(2,2),XNO(2),XOMEGO(2),TASCNP(2),XTO(2),  HECCOM
     2PERIOD(2),LEO,LOA,NREV1,NREV2,IYR,IYRP,IDAY,IPDAY,ICDAY,NREV1P, HECCOM
     3NREV2P,IV1DAY,IV2DAY,LYP                                        SET1
      COMMON /ACCEPH/                                                 ACCEPH
     1INNODE(17,2),SM(1,2),EC(1,2),XIN(1,2),RTASC(1,2),WP(1,2),TNDE(1,2)ACCEPH
     2,RG(1)                                                          ACCEPH
      COMMON /NREV/                                                   NREV
     1XLONG(1,2),NREV(2)                                              NREV
      DIMENSION NREVCT(1,2)                                           NREV
      EQUIVALENCE (NREV(1),NREVCT(1,1))                               NREV
      COMMON /SWAT/                                                   SWAT
     1DLONAN(100) ,ONMX(100),TIME1(100,3),DLONG1(100,3),INLAT(2),     SWAT
     2ISLAT(2),IS,STR,NEAT,LATNO(100),IALT(100),IGRDI(17)             SWAT
      COMMON /ACCNUM/ ACCNUM(100)                                     MAIN
      EQUIVALENCE(ACCNUM(1),KVEH),(ACCNUM(89),KDAY)                   MAIN
      EQUIVALENCE(IYEAR,IYR)                                          MAIN
      COMMON/ICARUS/ CARUS(8,17,2)                                    MAIN
      COMMON /XDAY/ IDAS(2)                                           SET7
      DATA CONS/10.E10/,CONS1/-10.E10/                                MAIN
      IFIRST = 1                                                      MAIN
      DO 5 I=1,1101                                                   MAIN
    5 CONTINUE                                                        MAIN
      NREV1 = 1                                                       MAIN
      NREV2 = 1                                                       MAIN
      CALL INPROC                                                     MAIN
      DO 15 I=1,6                                                     MAIN
      CARUS(I,1,1) = ORBIT1(I)                                        MAIN
      CARUS(I,1,2) = ORBIT2(I)                                        MAIN
   15 CONTINUE                                                        MAIN
      FLDA = IV1TIM(3)                                                DEBUG
      CALL IFPA(FLDA,IV1TIM(2),IV1TIM(1),ZZ,DAYS)                     DEBUG
      IDAY1 = DAYS                                                    MAIN
      ISTART = IDAY1                                                          *NEW
      XMU = CHU                                                               *NEW
      IF(IVEH .LT. 2)GO TO 17                                                 *NEW
      FLDA =IV2TIM(3)                                                 DEBUG
      CALL IFPA(FLDA,IV2TIM(2),IV2TIM(1),ZZ,DAYS)                     DEBUG
      IDAY2 = DAYS                                                    MAIN
      IF(IDAY1 .LE. IDAY2)ISTART = IDAY1                              MAIN
      IF(IDAY1 .GT. IDAY2)ISTART = IDAY2                              MAIN
   17 IEND = ISTART + NODAY - 1                                               *NEW
      IDAS(1) = IDAY1                                                 SET7    **-1
      IDAS(2) = IDAY2                                                 SET7
      DO 200 J=1,LEO                                                  MAIN
      DO 10 I=1,100                                                   SET3
```

```
      DLONMX(I) = CONS1                                           SET3
      DLONMN(I) = CONS                                            SET3
   10 CONTINUE                                                    SET3
      IF(J .EQ. 1)LYR = IV1TIM(1)                                 DEBUG
      IF(J .EQ. 2)LYR =IV2TIM(1)                                  DEBUG
      IDEND = NODAY                                               MAIN
      IF(J .GT. 1)GO TO 30                                        MAIN
      IYEAR = IV1TIM(1)                                           MAIN
      IYEAR = MOD(IYEAR,4)                                        MAIN
      ICDAY = 0                                                   MAIN
      CALL DAY(IV1TIM,IV1DAY)                                     MAIN
      IPDAY = IV1DAY - 2                                          MAIN
      IDAY = IV1DAY - 1                                           MAIN
      IYEARB = IV1TIM(1) + 1900 - 1                               MAIN
      IF(IDAY1 .GT. ISTART)IDEND = NODAY -(IDAY1 - ISTART)        MAIN
      GO TO 40                                                    MAIN
   30 IYEAR = IV2TIM(1)                                           MAIN
      IYEAR = MOD(IYEAR,4)                                        MAIN
      ICDAY = 0                                                   MAIN
      CALL DAY(IV2TIM,IV2DAY)                                     MAIN
      IPDAY = IV2DAY - 2                                          MAIN
      IDAY  = IV2DAY - 1                                          MAIN
      IYEARB = IV2TIM(1) + 1900 - 1                               MAIN
      IF(IDAY2 .GT. ISTART)IDEND = NODAY -(IDAY2 - ISTART)        MAIN
   40 IF(IDEND.GT. 0)GO TO 45                                     MAIN
      CALL PAGER(2)                                               DEBUG
      WRITE(KO,905)                                               DEBUG
      GO TO 500                                                   DEBUG
   45 RHO = CARUS(1,1,J   )*(1. -(CARUS(2,1,J   ))**2)            DEBUG     *NEW
      XG = 1. - 1.5*(SIN(CARUS(3,1,J   ))**2)                               *NEW
      YG = 1.5*CJ2*SQRT( 1. -(CARUS(2,1,J   ))**2)/RHO**2                   *NEW
      XK = 1. + YG*XG                                                       *NEW
      CMU = XMU*XK**2                                                       *NEW
      DO 170 I=1,IDEND                                                      *NEW
      ICDAY = ICDAY + 1                                           MAIN      **-1
      IDAY = IDAY + 1                                             MAIN
      IPDAY = IPDAY + 1                                           MAIN
      NREV1P = NREV1                                              MAIN
      NREV2P = NREV2                                              MAIN
      IF(MOD(IYEARB+IYEAR,4) .EQ. 0)GO TO 50                      MAIN
      IF(IDAY .LT. 366)GO TO 60                                   MAIN
      GO TO 55                                                    MAIN
   50 CONTINUE                                                    MAIN
      IF(IDAY .LT. 367)GO TO 60                                   MAIN
      IYEAR = MOD(IYEAR+1,4)                                      MAIN
      IDAY = IDAY - 366                                           DEBUG
      GO TO 60                                                    MAIN
   55 CONTINUE                                                    SET4
      IYEAR = MOD(IYEAR+1,4)                                      MAIN
      IDAY = IDAY - 365                                           MAIN.
   60 CONTINUE                                                    DEBUG
      IF(J .GT. 1)NREV(J)    = NREV2                              MAIN
      IF(J .EQ. 1)NREV(J)    = NREV1                              MAIN
      KDAY = I                                                    MAIN
      KVEH = J                                                    DEBUG
      CALL VECTOR                                                 DEBUG
      IF(KVEH .EQ. 1)NREV(KVEH) = NREV1                           MAIN
      IF(KVEH .GT. 1)NREV(KVEH) = NREV2                           SET2
      IF(IFIRST .EQ. 0)GO TO 70                                   SET2
      CALL OUTIS6(1)                                              MAIN
                                                                  MAIN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IFIRST = 0                                                          MAIN
   70 SH(1,J) = CARUS(1,1,J)                                              MAIN
      EC(1,J) = CARUS(2,1,J)                                              MAIN
      XTH(1,J) = CARUS(3,1,J)                                             MAIN
      RTASC(1,J) = CARUS(8,1,J)                                           DEBUG
      WP(1,J) = PIMOD(CARUS(5,1,J))                                       DEBUG
      TNOE(1,J) = TASCNP(J)                                               DEBUG
      RG(1) = PIMOD(CARUS(6,1,J) - CARUS(4,1,J) - WE*TASCNP(J))           DEBUG
   80 CONTINUE                                                            MAIN
      CALL ACCGEN                                                         MAIN
      DO 90 L=1,10,2                                                      MAIN
      IF(IPPTS(L) .LE. KDAY .AND. IPPTS(L+1) .GE. KDAY)GO TO 100          MAIN
   90 CONTINUE                                                            MAIN
      GO TO 170                                                           MAIN
  100 CALL PRTIS                                                          MAIN
  170 CONTINUE                                                            MAIN
      CALL WRITSW(3)                                                      MAIN
  200 CONTINUE                                                            MAIN
      CALL WRITSW(4)                                                      MAIN
      CALL EJECT(4)                                                       MAIN
      WRITE(KP,900)                                                       MAIN
  500 STOP                                                                DEBUG
  905 FORMAT(1H0,68H*** THE DIFFERENCE BETWEEN VEHICLE START DATES IS GR DEBUG
     1EATER THAN NODAY)                                                   DEBUG
  900 FORMAT(1H0,15X,39H*** SWATH GENERATION HAS BEEN COMPLETED)          SET2
      END                                                                 MAIN
```

```
@ ELT PAGER,1,760331, 59915    , 1
      SUBROUTINE PAGER(NLINES)                                           PAGER
C                                                                        PAGER
C                                                                        PAGER
      COMMON /CONS/                                                      CONS
     1ECCE,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,     CONS
     2INES,RE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,      CONS
     3ICASE,IPPTS(10),NLINE,NPAGE,INP,LINMAX,ISWTH,ISKK,NVEH,NODAY,      CONS
     4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDFBG(4)                            CONS
      REAL NAME,NAME1                                                    CONS
      LOGICAL LIDFBG                                                     CONS
      NLINE=NLINE+NLINES                                                 PAGER
      IF (NLINE.LT.LINMAX) GO TO 300                                     PAGER
      IF (NPAGE.EQ.0) GO TO 200                                          PAGER
      IMAX=LINMAX-NLINE+NLINES                                           PAGER
      GO TO 10                                                           PAGER
   10 CONTINUE                                                           PAGER
C                                                                        PAGER
C  SPACE TO BOTTOM OF PAGE                                               PAGER
C                                                                        PAGER
      IF (NLINES.LT.0) RETURN                                            PAGER
  200 CONTINUE                                                           PAGER
      NLINE=NLINES                                                       PAGER
      NPAGE=NPAGE+1                                                      PAGER
      WRITE (KO,9010)                                                             *NEW
      WRITE(KO,9020)(HEADER(I),I=1,18),ICASE,NPAGE                       PAGER    **-1
  300 CONTINUE                                                           PAGER
      RETURN                                                             PAGER
C                                                                        PAGER
C  FORMATS                                                               PAGER
C                                                                        PAGER
 9000 FORMAT (1H )                                                       PAGER
 9010 FORMAT (1H1)                                                                *NEW
 9020 FORMAT(1X,18A4,2X,16HLPP SIMULATION   ,5HCASE ,I4,2X,4HPAGE,I4)    PAGER    **-1
C                                                                        PAGER
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
@ ELT PIMOD,1,750925, 58784    , 1
      FUNCTION PIMOD(A)                                             PIMOD
C                                                                   PIMOD
C                                                                   PIMOD   **-5
C  GET POSITIVE ARGUMENT OF AN ANGLE IN RADIANS BETWEEN 0 AND 2PI   PIMOD
C                                                                   PIMOD
      B=6.2831853                                                   PIMOD
   10 IF(A) 20,50,30                                                PIMOD
   20 A=A+B                                                         PIMOD
      GO TO 10                                                      PIMOD
   30 IF(A-B) 50,50,40                                              PIMOD
   40 A=A-B                                                         PIMOD
      GO TO 30                                                      PIMOD
   50 PIMOD=A                                                       PIMOD
      RETURN                                                        PIMOD
      END                                                           PIMOD
```

ORIGINAL PAGE IS
OF POOR QUALITY

• FLT PRT15,1,750922, 69855   , 1

```
      SUBROUTINE PRTIS                                                  PRTIS
C     THIS ROUTINE PRINTS EPHEMERIS DATA FOR 1 DAY AND 1 VEHICLE        PRTIS
      COMMON /HFCCOM/                                                   HFCCOM
     1ALPHAT,XBETAD(2),XBETA(2,2),XWD(2),XOMEGD(2),TASCNP(2),XTO(2),    HFCCOM
     2PERIOD(2),LMO,LDA,NREV1,NREV2,IYR,IYRB,IDAY,IPDAY,ICDAY,NREV1P,   HFCCOM
     3NREV2P,IV1DAY,IV2DAY,LYP                                          SF11
      COMMON /CONS/                                                     CONS
     1ECCF,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),INOPT,ALPHAM,    CONS
     2THES,WE,ORBIT1(6),ORBIT2(6),PI,CJ2,CHU,PI2,PI32,SA(3),RADIAN,     CONS
     3ICASE,IPPI5(10),NLINE,NPAGE,INP,LINMAX,ISWTB,ISWR,NVEH,NODAY,     CONS
     4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDEBG(4)                           CONS
      REAL NAME,NAME1                                                   CONS
      LOGICAL LIDEBG                                                    CONS
      COMMON /FLAG/                                                     FLAG
     1IFIRST,IVEH,ISTART,IEND,IDEND                                     FLAG
      COMMON /ACCEPH/                                                   ACCEPH
     1TPNODE(17,2),SM(1,2),EC(1,2),XIN(1,2),RTASC(1,2),WP(1,2),TNDE(1,2)ACCEPH
     2,RG(1)                                                            ACCEPH
      COMMON /NREV/                                                     NREV
     1XLONG(1,2),NREV(2)                                                NREV
      DIMENSION NREVCT(1,2)                                             NREV
      EQUIVALENCE (NREV(1),NREVCT(1,1))                                 NREV
      COMMON/ICARDS/ CARDS(8,17,2)                                      PRTIS
      COMMON/ACCNUM/ ACCNUM(100)                                        PRTIS
      EQUIVALENCE(ACCNUM(7),PER(1)),(ACCNUM(85),DELNDE(1)),(ACCNUM(89), PRTIS
     1KDAY),(ACCNUM(1),KVEH)                                            SF11
      REAL PER(2),DELNDE(2)                                             PRTIS
      INTEGER XDEG(3,2),WDEG(3,2),RDEG(3,2),DEG(3)                      PRTIS
      JDAY = 1                                                          PRTIS
      CALL EJECT(14)                                                    PRTIS
      WRITE(KO,100)LMO,LDA,LYR,KVEH                                     SF14.
  100 FORMAT(/33X,42HT-5  PREDICTOR EPHEMERIS DISPLAY FOR DATE ,I2,1H/, SF13
     1I2,1H/,I2,//,44X                                                  SF13
     1,8HVEHICLE ,I1)                                                   PRTIS
      SMAT1 = SM(JDAY,KVEH)*RADIUS                                      PRTIS
      CALL DEGMOD(XIN(JDAY,KVEH),XDEG(1,KVEH))                          PRTIS
      CALL DEGMOD(WP(JDAY,KVEH),WDEG(1,KVEH))                           PRTIS
      CALL DEGMOD(RG(JDAY),RDEG(1,KVEH))                                PRTIS
      ECCPR = EC(JDAY,KVEH)                                             SF12
      PERPR = PER(KVEH)                                                 SF12
      WRITE(KO,150)SMAT1,ECCPR,PERPR,(XDEG(K,KVEH),K=1,3),(WDEG(K,KVEH),PRTIS
     1K=1,3),(RDEG(K,KVEH),K=1,3)                                       PRTIS
  150 FORMAT(/37X,21HSEMI-MAJOR AXIS      =, F10.3,                     PRTIS
     1/37X,21HECCENTRICITY         =,F11.8,                             PRTIS
     2/37X,21HPERIOD               =,F10.3,                             PRTIS
     3/37X,21HINCLINATION          =,I4,2I3,                            PRTIS
     4/37X,21HARGUEMENT OF PERIGEE=,I4,2I3,                             PRTIS
     5/37X,21HRT. ASC. GREENWICH   =,I4,2I3)                            PRTIS
      WRITE(KO,169)                                                     PRTIS
  169 FORMAT(/37X,31HREV   NODE TIME   NODE LONGITUDE)                  PRTIS
      MIN = NREV(KVEH)                                                  PRTIS
      DO 210 J=1,MIN                                                    PRTIS
      CALL DEGMOD(CARDS(4,J,KVEH),XDEG(1,KVEH))                         PRTIS
      CALL PAGER(1)                                                     PRTIS
      WRITE(KO,190)(J,TPNODE(J,KVEH),(XDEG(K,KVEH),K=1,3))              PRTIS
  190 FORMAT(37X,I2,1H),F12.4,3X,I4,2I3)                                PRTIS
  210 CONTINUE                                                          PRTIS
      RETURN                                                            PRTIS
```

PRTTS

END

000059

```
• ELT REVTAB,1,750925, 58796    , 1
      SUBROUTINE REVTAB                                                 REVTAB
C                                                                       REVTAB
C                                                                       REVTAB  *-11
C                                                                       REVTAB
C  SUBROUTINE REVTAB RECEIVES AS INPUT THE SWATH DATA TABLE FROM THE    REVTAB
C  SUBROUTINE SWATH. REVTAB REFORMATS THE TABLE TO OBTAIN THE DATA AT   REVTAB
C  EACH INTEGER VALUE OF LATITUDE.                                      REVTAB
C                                                                       REVTAB
C                                                                       REVTAB
       COMMON /CONS/                                                    CONS
      1ECCF,CIS,CHUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,   CONS
      2IHES,HE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,    CONS
      3ICASE,IPPTS(10),NLINE,NPAGE,INP,LINMAX,ISWTH,ISWR,NVEH,NODAY,    CONS
      4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDEBUG(4)                         CONS
       REAL NAME,NAME1                                                  CONS
       LOGICAL LIDEBUG                                                  CONS
       DIMENSION HO(5) ,XLONND(19,2)                                    REVTAB
       DIMENSION MAXLT(2),MINLT(2)                                      REVTAB
       COMMON/ACCNUM/ ACCNUM(100)                                       REVTAB
       EQUIVALENCE (ACCNUM(  1),KVEH)                                   REVTAB
      X,          (ACCNUM(  3),NUSCN)                                   REVTAB
      X,          (ACCNUM(  4),NADSCN)                                  REVTAB
      X,          (ACCNUM( 47),XLONND(1,1))                             REVTAB
      X,          (ACCNUM( 89),KDAY)                                    REVTAB
      X,          (ACCNUM( 90),F)                                       REVTAB
      X,          (ACCNUM( 92),MAXLT(1))                                REVTAB
      X,          (ACCNUM( 94),MINLT(1))                                REVTAB
      X,          (ACCNUM(  96),IADF)                                   REVTAB
C                                                                       REVTAB
C                                                                       REVTAB
       COMMON /ACCDAT/                                                  ACCDAT
      1TV(362),HV(362),SLAM(362,3),SPHI(362,3),ENP(2)                   ACCDAT
       COMMON /REVSTH/                                                  REVSTH
      1TIME(181,3,2),DLONG(181,3,2),ALT(181,2)                          REVSTH
       COMMON /ACCEPH/                                                  ACCEPH
      1TMNODE(17,2),SH(1,2),EC(1,2),XIN(1,2),RTASC(1,2),WP(1,2),TNDE(1,2)ACCEPH
      2,RG(1)                                                           ACCEPH
       COMMON /FLAG/                                                    FLAG
      1IFIRST,TVEH,ISTART,TEND,IDEND                                    FLAG
C                                                                       REVTAB
C                                                                       REVTAB
C                                                                       REVTAB
C  INITIALIZE OUTPUT ARRAYS FOR THIS VEHICLE                            REVTAB
C                                                                       REVTAB
       DO 10   I = 1,181                                                REVTAB
C                                                                       REVTAB
       ALT(I,KVEH) = 0.0                                                REVTAB
C                                                                       REVTAB
       DO 10   K = 1,3                                                  REVTAB
C                                                                       REVTAB
       DLONG(I,K,KVEH) = 2.E6                                           REVTAB
    10 TIME(I,K,KVEH) = 2.E6                                            REVTAB
C                                                                       REVTAB
C  DETERMINE ENTRY POINT IN SWATH DATA THAT CORRESPONDS TO N.P.         REVTAB
C                                                                       REVTAB
       TEMP = ENP(KVEH) - F                                             REVTAB
       IF (TEMP .LT. 0.0) TEMP = TEMP + TWOPI                           REVTAB
       NTRY = TEMP*RADIAN                                               REVTAB
```

```
C
 20   IF (TV(NTRY) .GT. TV(362))  GO TO 50
      NTRY = NTRY + 1
      GO TO 20
C
 50   IF (TV(NTRY) .LE. TV(362)) GO TO 100
      NTRY = NTRY - 1
      GO TO 50
C
C  INITIALIZE FOR DESCENDING PASS SWATH DATA
C
  100 CONTINUE
      TMXLAT = 0.
      XMXLAT = 0.
      TMNMX = 1.E6
      TMXMN = -100.
      TMNLAT = 0.
      XMNLAT = 0.
      JJ = 1
C  DETERMINE LATITUDE EXTREMES OF ACCESS TABLES
C
      DO 105 M = 1,NOSCN
      TEMP = SLAM(362,M)
      IF (TEMP .GT. 90.) GO TO 105
      TMNMX = AMIN1(TMNMX,TEMP)
      TMXLAT = AMAX1(TMXLAT,TEMP)
      IF (TEMP .GE. TMXLAT) GO TO 105
      XMXLAT = AMAX1(XMXLAT,TEMP)
  105 CONTINUE
C
  110 CONTINUE
      MXLAT = 91 - TMXLAT
      MNMX = 91 - TMNMX
C
      DO 115 M = 1,NOSCN
      TEMP = SLAM(361,M)
      IF (TEMP .GT. 90.) GO TO 115
      TMXMN = AMAX1(TMXMN,TEMP)
      TMNLAT = AMIN1(TMNLAT,TEMP)
      IF (TEMP .LE. TMNLAT) GO TO 115
      XMNLAT = AMIN1(XMNLAT,TEMP)
  115 CONTINUE
C
      MXMN = 91 - TMXMN
      MNLAT = 91 - TMNLAT
C
C
C  INITIALIZE POINTERS TO SWATH TABLE FOR EACH SCAN ANGLE
C
  125 DO 150  I = 1,3
  150 NO(I) = NTRY
      IF(.NOT. LIDEBG(4)) GO TO 160
      CALL PAGER(7)
      WRITE(KO,9030) MXLAT,MNMX,MXMN,MNLAT
      WRITE(KO,9040) (SLAM(361,K1),K1=1,NOSCN)
      WRITE(KO,9050) (SLAM(362,K1),K1=1,NOSCN)
      WRITE(KO,9025)
C
C  COMPUTE SCAN DATA FOR EACH LAT. ABOVE MINIMUM OF SWATH EDGES AT N.P.
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
 160  CONTINUE                                                            REVTAB
C                                                                         REVTAB
      DO 300  I  = MXLAT,MNMX                                             REVTAB
C                                                                         REVTAB
      XI  = 91 - I                                                        REVTAB
      INT = I                                                             REVTAB
      DO 250  K = 1,NOSCN                                                 REVTAB
C                                                                         REVTAB
      IF (SLAM(362,K) .LT. XI) GO TO 250                                  REVTAB
 175  L = NO(K)                                                           REVTAB
      IF (SLAM(L,K) .LE. XI) GO TO 200                                    REVTAB
      NO(K) = NO(K) + JJ                                                  REVTAB
      GO TO 175                                                           REVTAB
C                                                                         REVTAB
C  INTERPOLATE BETWEEN TWO VALUES FOR THIS SCAN ANGLE FROM TABLE TO       REVTAB
C  OBTAIN DATA FOR INTEGER LATITUDE                                       REVTAB
C                                                                         REVTAB
 200 IF( SLAM(L-JJ,K) .LE. 1.E6 .AND. SLAM(L,K) .NE. 1.E6) GO TO 210      REVTAB
      TIME(INT,K,KVEH) = 1.E6                                             REVTAB
      DLONG(INT,K,KVEH) = 1.E6                                            REVTAB
      GO TO 250                                                           REVTAB
 210 DEL = (XI-SLAM(L-JJ,K))/(SLAM(L,K)-SLAM(L-JJ,K))                     REVTAB
      TIME(INT,K,KVEH) = (TV(L)-TV(L-JJ))*DEL + TV(L-JJ)                  REVTAB
      DLONG(INT,K,KVEH) = (SPHI(L,K)-SPHI(L-JJ,K))*DEL + SPHI(L-JJ,K)     REVTAB
 250  CONTINUE                                                            REVTAB
C                                                                         REVTAB
      K = NADSCN                                                          REVTAB
      L = NO(NADSCN)                                                      REVTAB
      ALT(INT,KVEH) = ((XI-SLAM(L-JJ,K))/(SL   ..,K)-SLAM(L-JJ,K)))       REVTAB
     *                 * (HV(L)-HV(L-JJ)) + HV(L-JJ)                      REVTAB
 300  CONTINUE                                                            REVTAB
C                                                                         REVTAB
C                                                                         REVTAB
C  PROCESS INTEGER LATITUDES FROM LOWEST SWATH EDGE AT N.P. TO HIGHEST    REVTAB
C  SWATH EDGE AT S.P.                                                     REVTAB
C                                                                         REVTAB
C                                                                         REVTAB
 310  CONTINUE                                                            REVTAB
      I1 = MNMX + 1                                                       REVTAB
      I2 = MXMN - 1                                                       REVTAB
C                                                                         REVTAB
      DO 400 INT = I1,MXMN                                                REVTAB
C                                                                         REVTAB
C                                                                         REVTAB
      XI  = 91 - INT                                                      REVTAB
C                                                                         REVTAB
      DO 375  K = 1,NOSCN                                                 REVTAB
C                                                                         REVTAB
      IFL = 0                                                             REVTAB
 325  L = NO(K)                                                           REVTAB
      IF (SLAM(L,K) .LE. XI) GO TO 350                                    REVTAB
      NO(K) = NO(K) + JJ                                                  REVTAB
      IF ( NO(K) .GT. 0 .AND. NO(K) .LT. 361) GO TO 325                   REVTAB
      IF ( NO(K) .GT. 360) NO(K) = 1                                      REVTAB
      IF( NO(K) .LE. 0) NO(K) = 360                                       REVTAB
      IF ( IFL .GT. 0) GO TO 330                                          REVTAB
      IFL = 1                                                             REVTAB
      GO TO 325                                                           REVTAB
C                                                                         REVTAB
 330  CONTINUE                                                            REVTAB
```

```
      TIME(INT,K,KVEH) = 1.E6                                         REVTAB
      DLONG(INT,K,KVEH) = 1.E6                                        REVTAB
      GO TO 375                                                       REVTAB
C                                                                     REVTAB
C  INTERPOLATE BETWEEN TWO VALUES FOR THIS SCAN ANGLE FROM TABLE TO   REVTAB
C  OBTAIN DATA FOR INTEGER LATITUDE                                   REVTAB
C                                                                     REVTAB
  350 CONTINUE                                                        REVTAB
      AA = SLAM(L,K) - SLAM(L-JJ,K)                                   REVTAB
      IF( ABS(AA) .GT. 0.0) GO TO 360                                 REVTAB
      LLL = NO(NADSCN)                                                REVTAB
      CALL PAGER(2)                                                   REVTAB
      WRITE(KO,9060)                                                  REVTAB
      WRITE(KO,9070)K,L,AA,JJ,DEL,XI,INT,NTRY,LLL                     REVTAB
  360 IF( SLAM(L-JJ,K) .NE. 1.E6 .AND. SLAM(L,K) .NE. 1.E6) GO TO 365 REVTAB
      TIME(INT,K,KVEH) = 1.E6                                         REVTAB
      DLONG(INT,K,KVEH) = 1.E6                                        REVTAB
      GO TO 375                                                       REVTAB
  365 DEL = (XI-SLAM(L-JJ,K))/(SLAM(L,K)-SLAM(L-JJ,K))                REVTAB
      TIME(INT,K,KVEH) = (TV(L)-TV(L-JJ))*DEL + TV(L-JJ)              REVTAB
      DLONG(INT,K,KVEH) = (SPHI(L,K)-SPHI(L-JJ,K))*DEL + SPHI(L-JJ,K) REVTAB
  375 CONTINUE                                                        REVTAB
C                                                                     REVTAB
      K = NADSCN                                                      REVTAB
      L = NO(NADSCN)                                                  REVTAB
      AA = SLAM(L,K) - SLAM(L-JJ,K)                                   REVTAB
      IF ( ABS(AA) .GT. 0.0) GO TO 380                                REVTAB
      LLL = NO(NADSCN)                                                REVTAB
      CALL PAGER(2)                                                   REVTAB
      WRITE(KO,9060)                                                  REVTAB
      WRITE(KO,9070)K,L,AA,JJ,DEL,XI,INT,NTRY,LLL                     REVTAB
  380 CONTINUE                                                        REVTAB
      ALT(INT,KVEH) = ((XI-SLAM(L-JJ,K))/(SLAM(L,K)-SLAM(L-JJ,K)))    REVTAB
     *                * (HV(L)-HV(L-JJ)) + HV(L-JJ)                   REVTAB
  400 CONTINUE                                                        REVTAB
  406 CONTINUE                                                        REVTAB
C                                                                     REVTAB
C                                                                     REVTAB
C  PROCESS INTEGER LATITUDES FROM HIGHEST SWATH EDGE AT S.P. TO LOWEST REVTAB
C  SWATH EDGE AT S.P.                                                 REVTAB
C                                                                     REVTAB
C                                                                     REVTAB
C                                                                     REVTAB
C                                                                     REVTAB
      DO 500 INT = 12,MNLAT                                           REVTAB
C                                                                     REVTAB
C                                                                     REVTAB
      XI  =  91 - INT                                                 REVTAB
C                                                                     REVTAB
      DO 475  K = 1,NOSCN                                             REVTAB
C                                                                     REVTAB
      IF ( SLAM(361,K) .GT. XI) GO TO 475                             REVTAB
  425 L = NO(K)                                                       REVTAB
      IF (SLAM(L,K) .LE. XI) GO TO 450                                REVTAB
      NO(K) = NO(K) + JJ                                              REVTAB
      GO TO 425                                                       REVTAB
C                                                                     REVTAB
C  INTERPOLATE BETWEEN TWO VALUES FOR THIS SCAN ANGLE FROM TABLE TO   REVTAB
C  OBTAIN DATA FOR INTEGER LATITUDE                                   REVTAB
C                                                                     REVTAB
```

```
  450 IF( SLAM(L-JJ,K) .NE. 1.E6 .AND. SLAM(L,K) .NE. 1.E6) GO TO 465
      TIME(INT,K,KVEH) = 1.E6
      DLONG(INT,K,KVEH) = 1.E6
      GO TO 475
  465 DEL = (XI-SLAM(L-JJ,K))/(SLAM(L,K)-SLAM(L-JJ,K))
      TIME(INT,K,KVEH) = (TV(L)-TV(L-JJ))*DEL + TV(L-JJ)
      DLONG(INT,K,KVEH) = (SPHI(L,K)-SPHI(L-JJ,K))*DEL + SPHI(L-JJ,K)
  475 CONTINUE
C
      K = NADSCN
      L = NO(NADSCN)
      ALT(INT,KVEH) = ((XI-SLAM(L-JJ,K))/(SLAM(L,K)-SLAM(L-JJ,K)))
     *                 * (HV(L)-HV(L-JJ)) + HV(L-JJ)
  500 CONTINUE
  510 CONTINUE
C
C  TEST FOR ASCENDING SWATH DATA NOT REQUIRED OR ALREADY GENERATED
C
      IF ( .NOT. LIDENG(4)) GO TO 575
      CALL PAGER(1)
      WRITE(KO,9000)
      DO 550 MM = 1,181
      CALL PAGER(1)
      WRITE(KO,9005) TIME(MM,1,KVEH),DLONG(MM,1,KVEH),
     * TIME(MM,NADSCN,KVEH),DLONG(MM,NADSCN,KVEH),TIME(MM,NOSCN,KVEH),
     * DLONG(MM,NOSCN,KVEH),ALT(MM,KVEH),MM
  550 CONTINUE
  575 CONTINUE
C
C
C
  600 CONTINUE
C
C  SET MAXIMUM AND MINIMUM OF SWATH EDGES FOR TARGET PROCESSING
C
      MAXLT(KVEH) = 90 - MXLAT
      MINLT(KVEH) = 91 -MNLAT.
C
C
 9000 FORMAT(59H  REVTAB OUTPUT - TIME AND DELTA LONG FOR 3 POINTS, AND
     *ALT )
 9005 FORMAT( 7F14.7,I4)
 9010 FORMAT(21H 325/350 LOOP REVTAB ,6I6,2F15.7)
 9020 FORMAT(I5,8F14.7)
 9025 FORMAT(//)
 9030 FORMAT(//27H MXLAT, MNMX, MXMN, MNLAT =,4I8)
 9040 FORMAT(14H SLAM(361,L) =,8F14.8)
 9050 FORMAT(14H SLAM(362,L) =,8F14.8)
 9060 FORMAT(48H REVTAB ERROR - ANOMALY IN CROSS-TRACK LATITUDES)
 9070 FORMAT(1H ,2HK=,I2,1X,2HL=,I3,1X,3HAA=,F13.6,1X,3HJJ=,I1,1X,
     14HDEL=,F13.6,1X,3HXI=,E13.6,1X,4HINT=,I3,1X,5HNTRY=,I5,1X,
     24HLLL=,I3)
C
      RETURN
      END
```

```
@ FLT SWATH,1,751015, 4/511     , 1
      SUBROUTINE SWATH                                                  SWATH
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
C     SUBROUTINE SWATH UTILIZES THE EPHEMERIS DATA FOR THE DAY SPECIFIEDSWATH
C     TO GENERATE THE LATITUDE AND DELTA LONGITUDE DATA FOR THE SCAN    SWATH
C     ANGLES FOR ONE REV. THE DATA IS GENERATED AT ONE DEGREE INCREMENTSSWATH
C     OF ECCENTRIC ANOMALY, BEGINNING AT THE ASCENDING NODE.            SWATH
      COMMON /HECCOM/                                                   HECCOM
     1ALPHAT,XBETAD(2),XBETA(2,2),XWD(2),XOMEGD(2),TASCNP(2),X10(2),    HECCOM
     2PERIOD(2),LMO,LDA,NREV1,NREV2,IYR,IYRB,IDAY,IPDAY,ICDAY,NREV1P,   HECCOM
     3NREV2P,IV1DAY,IV2DAY,LYR                                          SETI
C                                                                       SWATH
      COMMON /CONS/                                                     CONS
     1ECCF,CIS,ENUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,    CONS
     2THES,WE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,     CONS
     3ICASE,IPPTS(10),NLINE,NPAGE,INP,LINMAX,ISWTH,ISWR,NVEH,NODAY,     CONS
     4IV1TIM(6),IV2TIM(6),RO,IORBIT,LIDFBG(4)                           CONS
      REAL NAME,NAME1                                                   CONS
      LOGICAL LIDFBG                                                    CONS
C                                                                       SWATH
      COMMON/ACCNUM/ACCNUM(100)                                         SWATH
      EQUIVALENCE (ACCNUM(  1),KVEH)                                    SWATH
     X,           (ACCNUM(  3),NOSCN)                                   SWATH
     X,           (ACCNUM(  4),NADSCN)                                  SWATH
     X,           (ACCNUM(  6),XN)                                      SWATH
     X,           (ACCNUM(  7),PER(1))                                  SWATH
     X,           (ACCNUM( 89),KDAY)                                    SWATH
     X,           (ACCNUM( 90),F)                                       SWATH
     X,           (ACCNUM( 91),TPER)                                    SWATH
C                                                                       SWATH
C                                                                       SWATH
      COMMON /ACCEPH/                                                   ACCEPH
     1IMNODE(17,2),SM(1,2),EC(1,2),XIN(1,2),RTASC(1,2),WP(1,2),TNDE(1,2)ACCEPH
     2,RG(1)                                                            ACCEPH
      COMMON /ACCDAT/                                                   ACCDAT
     1TV(362),HV(362),SLAM(362,3),SPHI(362,3),ENP(2)                    ACCDAT
      COMMON /FLAG/                                                     FLAG
     1IFIRST,IVEH,ISTART,IEND,IDEND                                     FLAG
      DIMENSION ZB(3),YB(3),XNOE(2),P(3),R(3),PER(2)                    SWATH
      DIMENSION XL(3)                                                   SWATH
C                                                                       SWATH
C                                                                       SWATH
C   COMPUTE TRUE ANOMALY OF NORTHERNMOST POINT ON REV                   SWATH
      JDAY = 1                                                          SWATH
C                                                                       SWATH
      WNP = FIMOD(TWOPI - WP(JDAY,KVEH) + PI2)                          SWATH
C                                                                       SWATH
C   SET CONSTANTS                                                       SWATH
C                                                                       SWATH
      ECC = EC(JDAY,KVEH)                                               SWATH
      TRM1 = 1. - ECC**2                                                SWATH
      TRM2 = SQRT(TRM1)                                                 SWATH
      TRM3 = TRM1/(1. + ECC*COS(WNP))                                   SWATH
C                                                                       SWATH
C   COMPUTE ECCENTRIC ANOMALY OF NORTHERNMOST POINT                     SWATH
C                                                                       SWATH
      SINE = TRM3*SIN(WNP)/TRM2                                         SWATH
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      COSF = (1. - TRM3)/ECC
      ENP(KVEH) = ACOS(COSF)
      IF (SINF .LT. 0.0) ENP(KVEH) = THOPI - ENP(KVEH)
C
C
C COMPUTE RIGHT ASCENSION OF GREENWICH AND LONG. OF NODE AT NODE TIME
C
      AG = PG(JDAY) +(WE-XOMEGD(KVEH))*TNDE(JDAY,KVEH)
      XLONGN = RTASC(JDAY,KVEH) - AG
      IF (XLONGN .LT. 0.0) XLONGN = XLONGN + THOPI
C
C
C
C
C
C
C COMPUTE POINTS AROUND REV
C
      IFL = 1
      DEL = 1./RADIAN
      EX  = E - DEL
      TPER = TPER - PER(KVEH)
C
      DO 400 I=1,362
C
C
      EX = EX + DEL
      IF(I .EQ. 361) GO TO 450
      IF(I .EQ. 362) GO TO 425
   40 TV(I) = (EX - ECC*SIN(EX))/XN + TPER
   45 IF (TV(I) .GE. TNDE(JDAY,KVEH)) GO TO 50
      TV(I) = TV(I) + PER(KVEH)
      GO TO 45
C
C COMPUTE TRUE ANOMALY AT THIS TIME POINT ON REV
C
   50 CONTINUE
      TV(I) = TV(I) - TNDE(JDAY,KVEH)
      TRM4 = 1. - ECC*COS(EX)
      SINV = TRM2*SIN(EX)/TRM4
      COSV = (COS(EX) - ECC)/TRM4
      V = ACOS(COSV)
      IF (SINV .LT. 0.0) V = THOPI - V
      TM = TV(I)
C
C COMPUTE INPLANE ANGLE FROM THE NODE AND VEHICLE ALTITUDE (E.R.)
C
      WT = V + WP(JDAY,KVEH)
      IF (WT .GT. THOPI) WT = WT - THOPI
      HV(I) = SP(JDAY,KVEH)*TRM1/(1.+ECC*COS(V)) - 1.
      HT = HV(I)
C
C VEHICLE GEOCENTRIC LATITUDE AND ECR Z COMPONENT OF POSITION
C
      ZV = SIN(WT)*SIN(XIN(JDAY,KVEH))
      VLAT = ASIN(ZV)
      SLA (I,NADSCN) = VLAT*RADIAN
C
C VEHICLE LONGITUDE AND ECR X,Y COMPONENTS OF POSITION
C
```

```
      AA = 0.                                                           SWATH
      IF (PI2 .LE. WT .AND. WT .LE. PI32) AA = 1.                       SWATH
      TEMP = XLONGN -(WE-XOMEGD(KVEH))*TV(I)                            SWATH
      IF ( I .LT. 361) GO TO 65                                         SWATH
      AA = 0.                                                           SWATH
      TEM = SIGN(PI2,PI2-XIN(JDAY,KVEH))                                SWATH
      IF ( I .EQ. 361) TEM = -TEM                                       SWATH
      GO TO 70                                                          SWATH
   65 TEM = ATAN(TAN(WT)*COS(XIN(JDAY,KVEH)))                           SWATH
   70 XLON = TEMP + TEM - AA*PI                                         SWATH
      XV = COS(VLAT)*COS(XLON)                                          SWATH
      YV = COS(VLAT)*SIN(XLON)                                          SWATH
      DXLON = XLON - XLONGN                                             SWATH
      SPHI(I,NADSCN) = DXLON                                            SWATH
      IF (SPHI(I,NADSCN) .GT. PI2) SPHI(I,NADSCN) = SPHI(I,NADSCN)      SWATH
     X  - THOPI                                                         SWATH
C                                                                       SWATH
C  Z BODY AXIS IS NEGATIVE OF VEHICLE POSITION VECTOR                   SWATH
C                                                                       SWATH
      ZB(1) = -XV                                                       SWATH
      ZB(2) = -YV                                                       SWATH
      ZB(3) = -ZV                                                       SWATH
C                                                                       SWATH
C  INSTANTANEOUS NODE VECTOR COMPONENTS (ECR)                           SWATH
C                                                                       SWATH
      XNDE(1) = COS(TEMP)                                               SWATH
      XNDE(2) = SIN(TEMP)                                               SWATH
C                                                                       SWATH
C  Y BODY AXIS NODE VECTOR CROSSED INTO Z-BODY AXIS                     SWATH
C                                                                       SWATH
      YB(1) = XNDE(2)*ZB(3)                                             SWATH
      YB(2) = -XNDE(1)*ZB(3)                                            SWATH
      YB(3) = XNDE(1)*ZB(2) - XNDE(2)*ZB(1)                             SWATH
C                                                                       SWATH
C  MAKE Y BODY AXIS A UNIT VECTOR AND CHANGE SIGN IF VEHICLE IS IN      SWATH
C  SOUTHERN HEMISPHERE                                                  SWATH
C                                                                       SWATH
      TEMP = SQRT(YB(1)**2 + YB(2)**2 + YB(3)**2)                       SWATH
      IF( VLAT .LT. 0.0) TEMP = -TEMP                                   SWATH
      DO 75   J = 1,3                                                   SWATH
   75 YB(J) = YB(J)/TEMP                                                SWATH
C                                                                       SWATH
C  COMPUTE VEHICLE POSITION VECTOR IN ECR COMPONENTS                    SWATH
C                                                                       SWATH
      D = 1. + HV(I)                                                    SWATH
      DO  100 J = 1,3                                                   SWATH
  100 P(J) = -D*ZB(J)                                                   SWATH
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
C                                                                       SWATH
C  COMPUTE SWATH DATA FOR EACH SCAN ANGLE                               SWATH
C                                                                       SWATH
      DO 300  K = 1,NOSCN                                               SWATH
C                                                                       SWATH
      IF (SA(K) .EQ. 0.0) GO TO 300                                     SWATH
      XLY = SIN(SA(K))                                                  SWATH
      XLZ = COS(SA(K))                                                  SWATH
```

```
C                                                                       SWATH
C  GENERATE COMPONENTS OF ECR DIRECTION VECTOR AT CURRENT SCAN ANGLE    SWATH
C                                                                       SWATH
      DO 150  L = 1,3                                                   SWATH
  150 XL(L) = XLY*YB(L) + XLZ*ZB(L)                                     SWATH
C                                                                       SWATH
C  GENERATE COMPONENT B (I.E.. DOT PRODUCT OF VECTORS XL AND P)         SWATH
C                                                                       SWATH
      XLMAG = SQRT(XL(1)**2 + XL(2)**2 + XL(3)**2)                      SWATH
      B = 0.0                                                           SWATH
      DO 200  L = 1,3                                                   SWATH
  200 B = B + XL(L)*P(L)                                                SWATH
      B = B/XLMAG                                                       SWATH
C                                                                       SWATH
C  COMPUTE DISTANCE S                                                   SWATH
C                                                                       SWATH
      IF ((B*B - D*D + 1.) .GE. 0.0) GO TO 225                          SWATH
C                                                                       SWATH
      GO TO 295                                                         SWATH
C                                                                       SWATH
C                                                                       SWATH
  225 CONTINUE                                                          SWATH
C                                                                       SWATH
      S = B + SQRT(B*B - D*D + 1.)                                      SWATH
C                                                                       SWATH
C  TEST FOR INTERSECTION OF THIS SCAN VECTOR WITH EARTH, IF NONE GO 300 SWATH
C                                                                       SWATH
      IF ( S .GT. 0.0) GO TO 295                                        SWATH
C                                                                       SWATH
C  COMPUTE ECR COMPONENTS OF VECTOR FROM EARTH CENTER TO INTERSECTION PTSWATH
C                                                                       SWATH
      DO 250  L = 1,3                                                   SWATH
  250 R(L) = P(L) - S*XL(L)/XLMAG                                       SWATH
C                                                                       SWATH
C  COMPUTE GEOCENTRIC LAT. AND DELTA LONG. FOR THIS INTERSECTION POINT  SWATH
C                                                                       SWATH
      IF (ABS(R(3)) .LE. 1.0) GO TO 275                                 SWATH
      RTEMP = SIGN(1.,R(3))                                             SWATH
      R(3) = RTEMP                                                      SWATH
C                                                                       SWATH
C                                                                       SWATH
  275 CONTINUE                                                          SWATH
      SLAM(I,K) = ASIN(R(3))*RADIAN                                     SET2
      TEMP = ATAN2(R(2),R(1))                                           SWATH
      TEM = PIMOD(TEMP) - XLONGN                                        SWATH
      IFLG = 0                                                          SWATH
  280 IF ( ABS(TEM-DXLON) .LT. PI2) GO TO 290                           SWATH
      IF ( IFLG .GT. 5) GO TO 288                                       SWATH
      IF (TEM .LT. 0.) GO TO 285                                        SWATH
      IFLG = IFLG + 1                                                   SWATH
      TEM = TEM - TWOPI                                                 SWATH
      GO TO 280                                                         SWATH
  285 TEM = TWOPI - TEM                                                 SWATH
      IFLG = IFLG + 1                                                   SWATH
      GO TO 280                                                         SWATH
  288 CALL FAFER(1)                                                     SWATH
      WRITE(KP,9010) XLONGN,TEM,DXLON,R(1),R(2),IFLG                    SWATH
  290 CONTINUE                                                          SWATH
      SPHI(I,K) = TEM                                                   SWATH
      GO TO 300                                                         SWATH
```

```
C                                                                        SWATH
C  ZERO LATITUDE AND LONGITUDE DIFFERENCE VALUES FOR THIS SCAN ANGLE     SWATH
C                                                                        SWATH
 295  CONTINUE                                                           SWATH
      SLAM(I,K) = 1.0E6                                                  SWATH
      SPHI(I,K) = 1.0E6                                                  SWATH
C                                                                        SWATH
 300  CONTINUE                                                           SWATH
C                                                                        SWATH    **-1
C  COMPUTE DATA AT N. P. AND S. P.                                       SWATH
C                                                                        SWATH
                                                                                  *NEW
      GO TO 400
 425  CONTINUE                                                           SWATH    **-1
      EX = FNP(KVEH)                                                     SWATH
      IFL = 2                                                            SWATH
      GO TO 40                                                           SWATH
 450  CONTINUE                                                           SWATH
      WSP = WNP + PI                                                     SWATH
      TRM3 = TRM1/(1. + ECC*COS(WSP))                                    SWATH
      SINE = TRM3*SIN(WSP)/TRM2                                          SWATH
      COSE = (1. - TRM3)/ECC                                             SWATH
      EX = ACOS(COSE)                                                    SWATH
      IF ( SINE .LT. 0.0) EX = TWOPI - EX                                SWATH
      IFL = 3                                                            SWATH
      GO TO 40                                                           SWATH
                                                                                  *NEW
 400  CONTINUE
C                                                                        SWATH
C                                                                        SWATH
C                                                                        SWATH
 600  CONTINUE                                                           SWATH
      IF ( .NOT. LIDEBG(1)) GO TO 620                                    SWATH
C                                                                        SWATH
      NADSC = NADSCN-1                                                   SWATH
      CALL EJECT(2)                                                      SWATH
      WRITE( 6,9000)                                                     SWATH
      DO 610 I=1,362                                                     SWATH
      CALL PAGER(1)                                                      SWATH
      WRITE(KO,9005)((SLAM(I,J),SPHI(I,J),J=1,NOSCN),HV(I),TV(I),I)      SF12
 610  CONTINUE                                                           SWATH
C                                                                        SWATH
C                                                                        SWATH
 9000 FORMAT(62H SWATH OUTPUT- LAT. AND DELTA LONG FOR 3 POINTS, ALT ANDSWATH
     X TIME )                                                            SWATH
 9005 FORMAT( 8F14.7,I4)                                                 SWATH
 9010 FORMAT(26H ITERATION LIMIT ON SWATH ,5F15.8,I3)                    SWATH
 620  RETURN                                                             SWATH
      END                                                                SWATH
```

```
* FLT WRITSW,1,750925, 58804   , 1

      SUBROUTINE WRITSW(IOPT)                                          WRITSW
C     THIS ROUTINE INITIALIZES,WRITES AND TERMINATES OUTPUT RECORDS ON WRITSW
C     THE SWATH TABLE AN SWATH REFERENCE FILES.                        WRITSW
C     NOTE THIS ROUTINE MUST BE RECODED FOR UNIVAC AND THE PDP.        WRITSW
      COMMON /CONS/                                                    CONS
     1ECCE,CIS,CNUV,RADIUS,HEADER(20),NAME(2),NAME1(2),TWOPI,ALPHAM,   CONS
     2THES,WE,ORBIT1(6),ORBIT2(6),PI,CJ2,CMU,PI2,PI32,SA(3),RADIAN,    CONS
     3ICASE,IPRTS(10),NLINE,NPAGE,INP,LINMAX,ISWIR,ISWR,NVEH,NUDAY,    CONS
     4IV1TIM(6),IV2TIM(6),KO,IORBIT,LIDEBG(4)                          CONS
      REAL NAME,NAME1                                                  CONS
      LOGICAL LIDEBG                                                   CONS
      COMMON /HECCOM/                                                  HECCOM
     1ALPHAT,XBETAD(2),XBETA(2,2),XHD(2),XOMEGD(2),TASCNP(2),XTO(2),   HECCOM
     2PERIOD(2),LMU,LDA,NREV1,NREV2,IYR,IYRR,IDAY,IFDAY,ICDAY,NREV1P,  HECCOM
     3NREV2P,IV1DAY,IV2DAY,LYR                                         SET1
      COMMON /ACCEPH/                                                  ACCEPH
     1TIMEOF(17,2),SM(1,2),EC(1,2),XIN(1,2),RTASC(1,2),WP(1,2),TNDE(1,2)ACCEPH
     2,RG(1)                                                           ACCEPH
      COMMON /NREV/                                                    NREV
     1XLONG(1,2),NREV(2)                                               NREV
      DIMENSION NREVCT(1,2)                                            NREV
      EQUIVALENCE (NREV(1),NREVCT(1,1))                                NREV
      COMMON /SWAT/                                                    SWAT
     1DLONMN(100),DLONMX(100),TIME1(100,3),DLONG1(100,3),INLAT(2),     SWAT
     2ISLAT(2),ISOSTR,NLAT,LATNO(100),IALT(100),ICRDI(17)              SWAT
      COMMON/ACCNUM/ ACCNUM(100)                                       WRITSW
      EQUIVALENCE (ACCNUM(1),KVEH),(ACCNUM(89),KDAY)                   WRITSW
      COMMON/ICARDS/ CARDS(8,17,2)                                     WRITSW
      COMMON /FLAG/                                                    FLAG
     1IFIRST,IVEH,ISTART,IEND,IDEND                                    FLAG
      COMMON /XDAY/ IDAS(2)                                            SET7.
      DIMENSION A(800),XNLAT(2),XSLAT(2),V1TIM(3),V2TIM(3)             WRITSW
      DIMENSION ALTI(100),ATNO(100)                                    WRITSW
      EQUIVALENCE (ALTI(1),IALT(1)),(ATNO(1),LATNO(1))                 WRITSW
      EQUIVALENCE(XCASE,ICASE),(XVEH,NVEH),(XNUDAY,NUDAY),( XLAT,NLAT),WRITSW
     1(XNLAT(1),INLAT(1)),(XSLAT(1),ISLAT(1)),(V1TIM(1),IV1TIM(1)),    WRITSW
     2(V2TIM(1),IV2TIM(1)),(SOSTR,ISOSTR)                              WRITSW
      DO 5 I=1,800                                                     WRITSW
      A(I) = 0.                                                        WRITSW
    5 CONTINUE                                                         WRITSW
      IF(IOPT .EQ. 2)GO TO 100                                         WRITSW
      IF(IOPT .EQ. 3)GO TO 200                                         WRITSW
      IF(IOPT .EQ. 4)GO TO 300                                         WRITSW
      DEFINE FILE 8(1101,800,U,IDUM)
      A(1) = NAME(1)                                                   WRITSW
      A(2) = NAME(2)                                                   WRITSW
      A(3) = XCASE                                                     WRITSW
      A(4) = XVEH                                                      WRITSW
      A(5) = XNUDAY                                                    WRITSW
      A(7) = XNLAT(1)                                                  WRITSW
      A(8) = XNLAT(2)                                                  WRITSW
      A(9) = XSLAT(1)                                                  WRITSW
      A(10)= XSLAT(2)                                                  WRITSW
      DO 8 I=1,3                                                       WRITSW
      K = I + 10                                                       WRITSW
      L = I +13                                                        WRITSW
      A(K) =  V1TIM(I)                                                 WRITSW
      A(L) =  V2TIM(I)                                                 WRITSW
```

```
    8 CONTINUE                                                          WRITSW
      A(17) = SOSTR                                                     WRITSW
      A(6) = XLAT                                                       WRITSW
      NOWDS = 8*NLAT                                                    WRITSW
      FIND(ISWTB'1)
      WRITE(ISWTB'1,ERR=600) (A(I),I=1,NOWDS)
      REWIND ISWR                                                       WRITSW
      WRITE(ISWR) NAME1(1),NAME1(2),ICASE,NVEH,NODAY,(IV1TIM(I),I=1,3), WRITSW
     1(IV2TIM(I),I=1,3),((CARUS(I,J),J=1,2),I=1,3),XOMEGO(1),XOMEGO(2)WRITSW
     2,XWD(1),XWD(2),(SA(I),I=1,3),(A(I),I=23,54)                       SET3
      GO TO 500                                                         WRITSW
  100 IS1 = NLAT                                                        WRITSW
      IS2 = 2*NLAT                                                      WRITSW
      IS3 = 5*NLAT                                                      WRITSW
      DO 110 I=1,NLAT                                                   WRITSW
      A(I) =  ATNO(I)                                                   WRITSW
      A(IS1+I) = ALTI(I)                                                WRITSW
  110 CONTINUE                                                          WRITSW
      DO 130 J=1,3                                                      WRITSW
      IPT = (J-1)*NLAT                                                  WRITSW
      DO 120 I=1,NLAT                                                   WRITSW
      A(IS2+IPT+I) = TIME1(I,J)                                         WRITSW
      A(IS3+IPT+I) = DLONG1(I,J)                                        WRITSW
  120 CONTINUE                                                          WRITSW
  130 CONTINUE                                                          WRITSW
      IHRS = TEND - IDAS(1) + 1                                         SET7
      IRLEN = (IHRS +1)*(KVEH-1) + KDAY + 1                             SET7
      FIND(ISWTB'IRLEN)
      WRITE(ISWTB'IRLEN,ERR=600) (A(I),I=1,NOWDS)
      WRITE(ISWR)KVEH,KDAY,WP(1,KVEH),NREV(KVEH),(TMNODE(I,KVEH),I=1,17)SET1
     1,                                                                 SET1
     1(CARUS(4,I,KVEH),I=1,17),(IGRD1(I),I=1,17)                        WRITSW
      IF(.NOT.LIDEBG(2))GO TO 150                                       SET1
      NR = NREV(KVEH)                                                   SET2
      CALL PAGER(NR+4)                                                  SET2
      WRITE(KO,900)KVEH,KDAY                                            SET1
  900 FORMAT(1H0,27X,32HSWATH REFERENCE RECORD   VEHICLE,1X,I1,3X,3HDAY,SET1
     11X,I3)                                                            SET2
      WRITE(KO,901)                                                     SET1
  901 FORMAT(1H0,33X,38HWP         NREV        TMNODE          CARUS)   SET2
      WRITE(KO,902) WP(1,KVEH),((I,TMNODE(I,KVEH),CARUS(4,I,KVEH)),I=1, SET1
     1NR)                                                               SET1
  902 FORMAT(1H ,27X,F13.6,3X,I2,4X,E12.6,2X,F13.6//,(1H ,43X,I2,4X,    SET3
     1E12.6,2X,F13.6/))                                                 SET3
      CALL PAGER(NLAT+4)                                                SET2
      WRITE(KO,903)KVEH,KDAY                                            SET1
  903 FORMAT(1H0,27X,28HSWATH TABLE RECORD   VEHICLE,1X,I1,3X,3HDAY,1X, SET1
     1I3)                                                               SET2
      WRITE(KO,904)                                                     SET1
  904 FORMAT(1H0,13X,80H LATNO  IALT  TIME(1)  TIME(2)  TIME(3)    DLONGSET2
     11(1)      DLONG1(2)     DLONG1(3))                                SET2
      WRITE(KO,905)((LATNO(I),IALT(I),(TIME1(I,J),J=1,3),(DLONG1(I,J),  SET1
     1J=1,3)),I=1,NLAT)                                                 SET1
  905 FORMAT(1H ,15X,I3,2X,I5,3X,F5.0,4X,F5.0,4X,F5.0,3X,                         *NEW
     1  E12.6,2X,E12.6,2X,E12.6)                                                  *NEW
  150 CONTINUE                                                          SET1      **-1
      GO TO 500                                                         WRITSW
  200 DO 210 I=1,NLAT                                                   WRITSW
      A(I) = ATNO(I)                                                    SET7
      A(I+NLAT) = DLONNN(I)                                             SET3
```

```
      A(I+2*NLAT) = DLONMX(1)                                           SET3
  210 CONTINUE                                                          WRITSW
      IWR1 = IEND - IDAS(1) + 1                                         SF17
      IWRS = IEND - IDAS(KVEH) + 1                                      SF17
      IRECN = (IWR1+1)*(KVEH-1) + IWR3 + 2                              SF17
      FIND(ISWTB'IRECN)
      WRITE(ISWTB'IRECN,ERR=600) (A(I),I=1,NOWDS)                       SF12
      IF(.NOT. IDEBG(2))GO TO 500                                       SF12
      CALL PAGER(NLAT+4)                                                SF12
      WRITE(KO,910)KVEH                                                 SF12
  910 FORMAT(1H0,27X,34HSWATH TABLE SEARCH RECORD VEHICLE ,1X,I1)       SF12
      WRITE(KO,915)                                                     SF12
  915 FORMAT(1H0,27X,37HLATNO      DLONMX                 DLONMN)       SF12
  920 FORMAT(1H ,20X,I5,3X,F13.6,9X,E15.6)                              SF12
      WRITE(KO,920)(LATNO(I),DLONMX(I),DLONMN(I),I=1,NLAT)              WRITSW
      GO TO 500
  300 CONTINUE                                                          WRITSW
      END FILE ISWR                                                     WRITSW
      REWIND ISWR                                                       WRITSW
  500 RETURN
  600 WRITE(KO,930)
  930 FORMAT(1H0,83H *** AN IRRECOVERABLE I/O ERROR HAS OCCURRED ON WRIT
     1ING A RECORD ON THE SWATH TABLE   /10X,31H THE JOB IS BEING ABANDO
     2NED ***)
      STOP                                                              WRITSW
      END
```

3. TOC

GRID BOOK II

Table of Contents

PART I

PROBLEM DESCRIPTION

UTILITY PROGRAM GRID

1.0 SCOPE

1.1 PROGRAM CAPABILITIES

This utility processes NASA's climatology tape to obtain the weather index along with its associated latitude and longitude. It then converts the latitude and longitude to a grid no pair (I, J). Using J as a record number, the utility writes a record onto an index matrix location file. Each record contains all the weather indexes associated with a given J value.

1.2 OPERATIONAL ASSUMPTIONS

- The weather index is in the range from 1 to 16000.

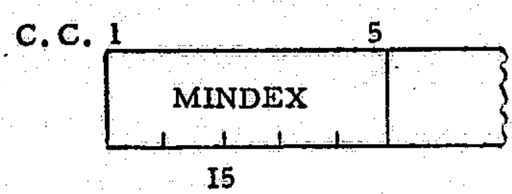
2.0 INPUT

2.1 CARDS

2.1.1 Data Quantities

MINDEX - maximum number of index points to process on the NASA climatology tape. Range 1 to 16000, nominal value is 16000 (used to reduce run time for debugging cases).

2.1.2 Card Format

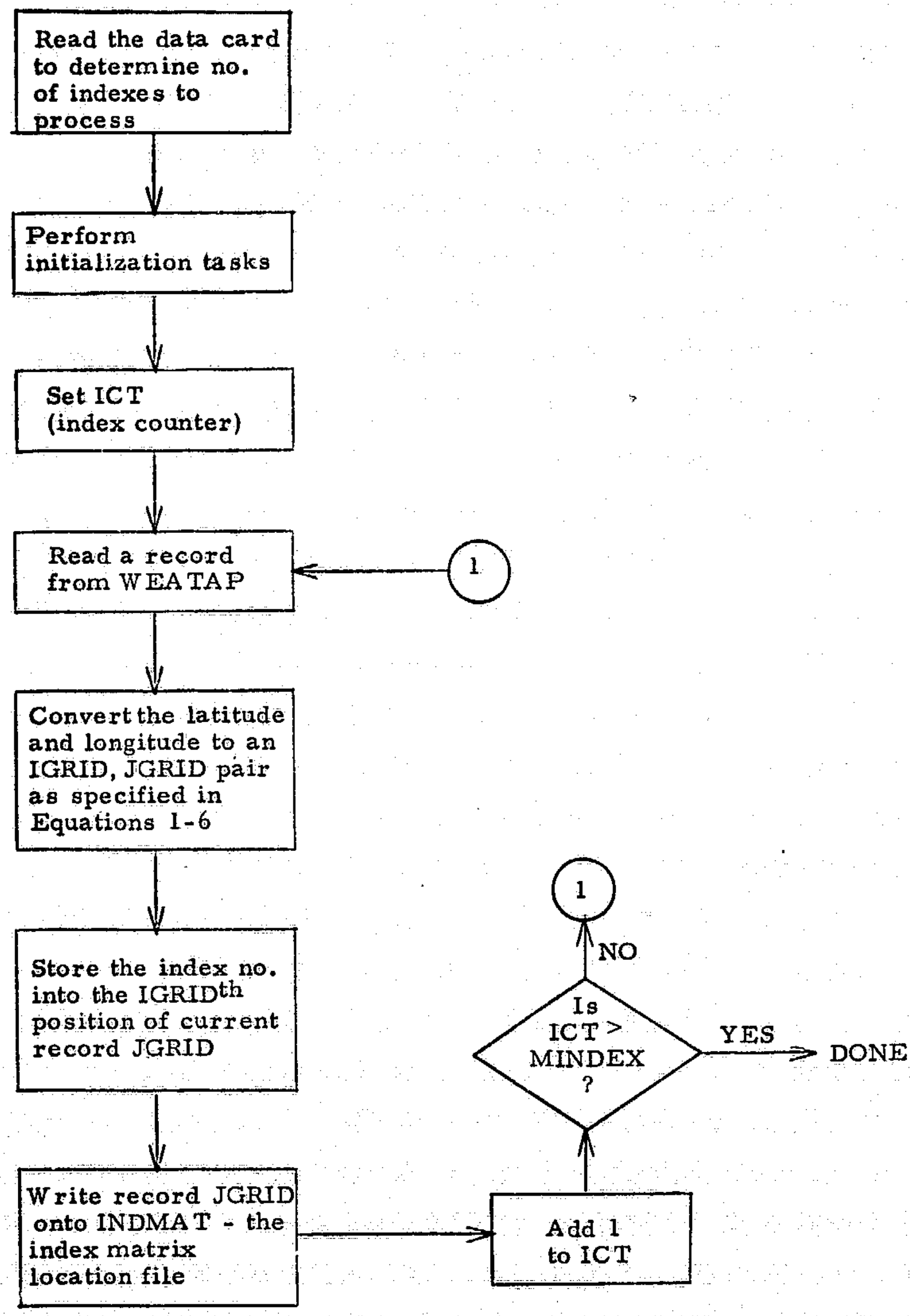


2.2 FILES

The only input file is the NASA climatology tape assigned to WEATAP - unit 13. See Section 2.4 of the Users Manual for a description of the file.

3.0 PROCESSING

3.1 PROGRAM FLOW



3.2 EQUATIONS

Given ϕ - latitude in radians

λ - longitude in radians

$I\phi = 125$, $J\phi = 125$ - North Pole index set

Compute stereographic projection

1. $\phi' = ABS(\phi)$

2. $R = 124.817494012 * \tan\left(\frac{\pi - 2 * \phi'}{4}\right)$

3. $A = \lambda - \pi/18$

4. $FI = I\phi + R * \cos(A)$
   $FJ = J\phi - R * \sin(A)$

5. If ϕ is minus $FI = FI + 2 * I\phi$

6. $IGRID = FI + .5$
   $JGRID = FJ + .5$

IGRID - the horizontal axis grid coordinate

JGRID - the vertical axis grid coordinate

4.0 OUTPUT

4.1 PRINT DATA

There is no normal printed output.

4.2 FILES

The only output file is INDMAT - the index matrix location file. See Section 2.4 of the Users Manual for a description of its contents.

5.0 ERROR PROCESSING

5.1 GENERAL

There are only three checks made for errors, one of which is fatal.

5.2 ERROR CHECKS AND MESSAGES

Fatal

1. A check is made to make sure MINDEX on the input card is between 0 and 16000. If not, the following message prints and the program stops.

   Message:

   MINDEX IS NOT BETWEEN 0 AND 16000. PROGRAM IS TERMINATED.

Non-Fatal

2. A check is made to make sure the index no. read from WEATAP file is between 1 and 16000. If not, the index no. is not included in the current INDMAT file record and the program reads the next WEATAP record.

   Message:

   THE INDEX NO. READ FROM WEATAP IS NOT BETWEEN 1 AND 16000. THE DATA IS IGNORED.

3. A check is made to see if an IGRID, JGRID grid pair has already been assigned an index number. If so, a message is printed and the current index no. replaces the one originally stored.

   Message:

   THE GRID PAIR IGRID nnn JGRID nnn HAS AN INDEX NUMBER ALREADY ASSIGNED.

PART II

COMMON BLOCK DEFINITIONS

COMMON STORAGE ALLOCATION

Name DIRAC Size ____ Page ____ of ____

Function Contains data for INDMAT direct access file

Name	Dimension	Format	Description	Symbol	Units
ICDCIN	252	I	Index array for CDC to support INDMAT file		
IREC	500	I	One record for INDMAT file current record to be read/written		
IGRID	1	I	The position in IREC in which to store IGRPT (row no.)		
JGRID	1	I	The current column ≡ rec. no. to read or write		
JSAVE	1	I	The previous value of JPT		
MAXJ	1		= 41 for hand off case test = 251 for final version		

COMMON STORAGE ALLOCATION

Name WEATDT Size______ Page_____ of_____

Function Contains data needed from weather tape

Name	Dimension	Format	Description	Symbol	Units
IGRDPT	1	I	Index no. - 1 to 16000		
LAT	4	I, A	LAT(1) + deg, LAT(2) - min LAT(3) - sec, LAT(4) - S or N		
LON	4	I, A	LON(1) - deg, LON(2) - min LON(3) - sec, LON(4) - E or W		
IA	(500)	I	Weather tape input record area - physical		
IB	(50)	I	One logical UNIVAC tape record		
IC	(30)	I	One logical CDC tape record		

COMMON STORAGE ALLOCATION

Name FLAG Size ________ Page _____ of _____

Function ______________________________

Name	Dimension	Format	Description	Symbol	Units
INP	1	I	Input card unit ≡ 5		
IOUT	1	I	Print unit ≡ 6		
IWTAPE	1	I	Weather tape unit ≡ 13		
IGRDTP	1	I	Grid matrix file unit ≡ 4		

PART III

LIST OF SUBROUTINES AND SUBROUTINE CALL STRUCTURE

LIST OF SUBROUTINES IN GRID

	Name	Function
1.	GRID	Main control routine for GRID program. It reads the input control card, controls the reading of the weather tape, computes latitude and longitude and controls the writing of the index matrix file.
2.	RDGRID	Reads one record from the index matrix file.
3.	WRGRID	Writes the index matrix file.
4.	RDWETR	Reads the NASA weather tape.

SUBROUTINE CALL STRUCTURE

GRID

- WRGRID
- RDWETR
- RDGRID

PART IV

SUBROUTINE DESCRIPTION

AND

FLOWCHARTS

SUBROUTINE GRID

Purpose:

This is the main control routine for the GRID program. It reads the control card, controls the reading of the weather tape, computes from the latitude and longitude associated with a weather grid the I, J coordinates of the index matrix and controls the writing of the index matrix file.

Input:

The quantity MINDEX is obtained from the control card.

IGROPT, LAT and LON are obtained from the file IWTAPE.
INP and IOUT from FLAG COMMON.

Output:

MAXJ records on file IGRDTP.

Linkage:

None.

Subroutines Used:

CALL RDWETR(IRC) - IRC is relative logical Rec. No. within a physical record. Initialized to 1 first time only.

CALL RDGRID - Read 1 record JGRID from IGRDTP file.

CALL WRGRID(IOP) - IOP = 0. Open IGRDTP file and initialize all MAXJ rec. to 0.

IOP = +N. Write record from IREC to IGRDTP file.

IOP = -N. Write record and close the IGRDTP file.

Local Variables:

INDEX - Count on no. of weather index rec. to read from IWTAPE

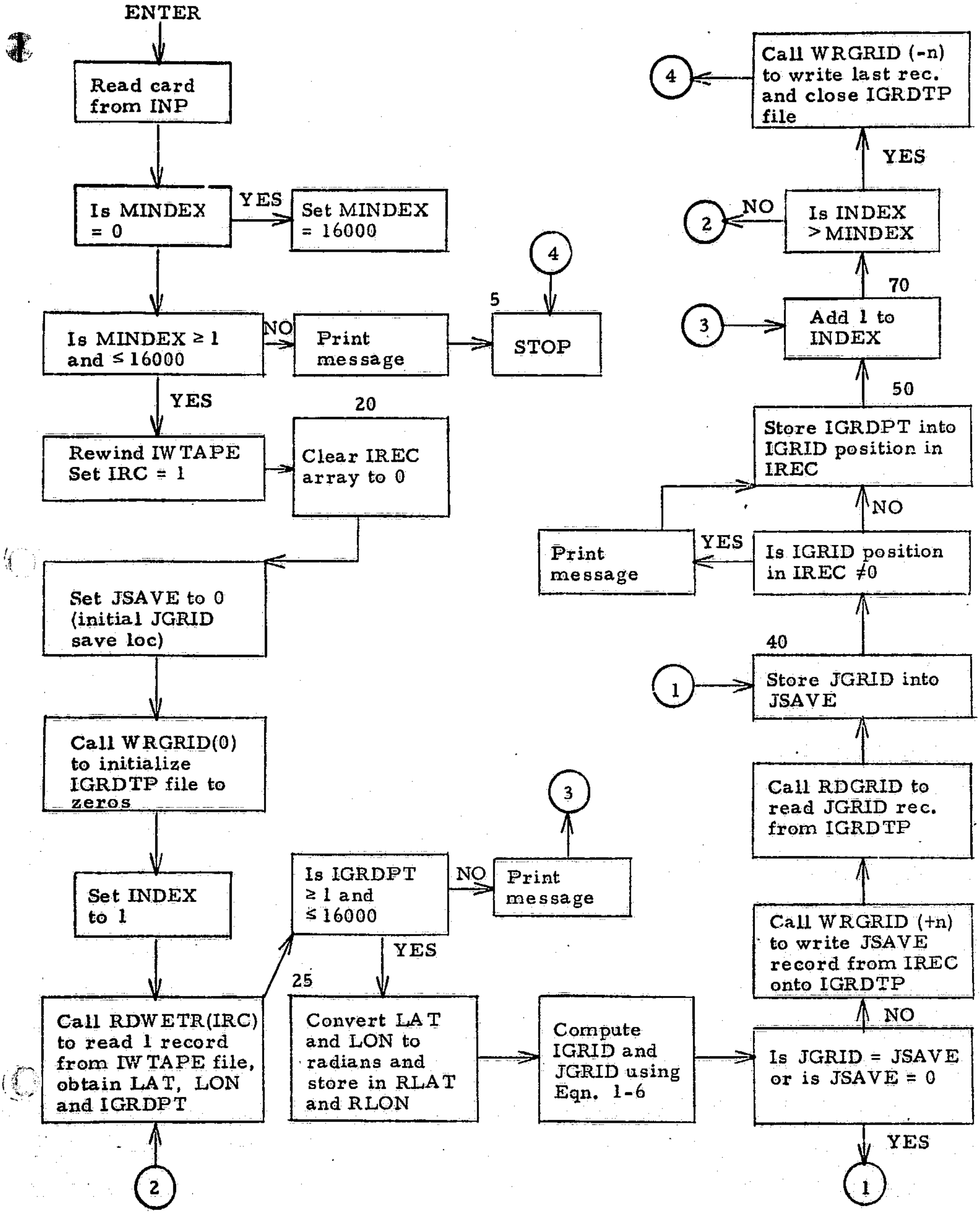
RLAT - Latitude in radians

RLON - Longitude in radians

Processing:

See flowchart for logic flow.
Equations 1-6 are described in the Problem Description.

Subroutine GRID Block Diagram



SUBROUTINE RDGRID

Purpose:

To read one record from the direct access file IGRDTP. The Record No. to be read is stored in JGRID and the record is read into IREC. For the CDC, READMS is to be used and for the UNIVAC, the direct access read. The record length is 500 words.

Input:

ICDCIN, JGRID in DIRAC COMMON.
IGRDTP in FLAG COMMON.
The data record on IGRDTP file.

Linkage:

CALL RDGRID.

Subroutines Used:

None.

Processing:

Give a call to READMS (CPC) or read from unit IGRDTP for UNIVAC.

SUBROUTINE WRGRID

Purpose:

On option, create a NULL IGRDTP D.A. file, or write one record at the JSAVE location from IREC or write a record and close the file. The record length is 500 words.

Input:

ICDCIN, IREC, JSAVE, MAXJ, JGRID in DIRAC COMMON.
IGRDTP in FLAG COMMON.

Output:

Either a NULL file on IGRDTP or one record in the JSAVE loc. of IGRDTP from IREC.

Linkage:

CALL WRGRID(IOP)

IOP = 0 - Generate NULL file.
IOP = +n - Write one record.
IOP = -n - Write one record and close the file.

Subroutines Used:

None.

Processing:

IOP = 0 - Give call to OPENMS for CDC and define file for UNIVAC (MAXJ gives no. of records), and then write MAXJ records from IREC using WRITEMS or WRITE (IREC has been cleared to zero).

IOP = +n - Give call to WRITEMS or write to WRITE IREC on JSAVE record.

IOP = -n - Give call to WRITEMS or write to WRITE IREC on JGRID record. Give call to CLOSEMS or call CLOSE (IGRDTP, 0) to close the file.

SUBROUTINE RDWETR

Purpose:

This routine reads one physical record from the NASA weather tape. It unblocks the record and stores one logical record into working storage.

Input:

One physical record on IWTAPE unit.

COMMON FLAG
  IWTAPE

COMMON WEATDT
  IA, IB

Output:

COMMON WEATDT
  IGRDPT, LAT, LON

Linkage:

CALL RDGRID(IRC). IRC is a logical record count 1 to 10 initialized to 1 by the calling routine for first call only. Routine then adds 1 to it and resets it to 1 as needed.

Method/Procedure:

Use NTRAN to perform the read from the weather tape.

Subroutines Called:

NTRAN.

PART V

SUBROUTINE LISTINGS

@ FLT BLKDAT,1,760401, 68088 , 1

```
      BLOCK DATA                                                        BLOCK
      COMMON /FLAG/                                                     FLAG
     1INP,IOUT,IHTAPE,IGRDTP                                            FLAG
      COMMON /DTRAC/                                                    DTRAC
     1ICDCIN(252),IREC(500),IGRTD,JGRTD,JSAVE,MAXJ                      DEBUG1
      COMMON /WEATDT/                                                   WEATDT
     1IGRDPT,LAT(4),LON(4),TA(500),IB(50 ),IC(50)
      DATA MAXJ/42/,INP/5/,IOUT/6/,IHTAPE/13/,IGRDTP/4/                 DEBUG1
      END                                                               BLOCK
```

ORIGINAL PAGE IS OF POOR QUALITY

```
• ELT GRID•1•760401• 68090     • 1

C      PROGRAM GRID(ITEST=401,OUTPUT=401,TAPE5=ITEST,TAPE6=OUTPUT,        DEBUG1
C     1INDMAT=1004,WEATAP=1004,TAPE4=INDMAT,TAPE13=WEATAP)                GRID
       COMMON /FLAG/                                                      FLAG
      1INP,IOUT,IWTAPE,IGRDTP                                             FLAG
       COMMON /WEATDT/                                                    WEATDT
      1IGROPT,LAT(4),LON(4),IA(500),IU(50 ),IC(30)
       COMMON /DTRAC/                                                     DTRAC
      1IRDCIN(252),IREC(500),IGRID,JGRID,JSAVE,MAXJ                       DEBUG1
       DIMENSION SLAT(3),SLON(3)                                          GRID
       DATA DEGTOR/.017453292/,IO/125/,JO/125/,PI/3.1415926/              GRID
       READ(INP,900)MINDEX                                                GRID
       IF(MINDEX .EQ. 0)MINDEX = 16000                                    GRID
       IF(MINDEX .GE. 1 .AND.MINDEX .LE. 16000)GO TO 10                   GRID
       WRITE(IOUT,901)                                                    GRID
     5 STOP                                                               GRID
    10 CALL NTRAN (IWTAPE, 10, 22)                                        GRID
       IRC = 1                                                            GRID
       DO 20 I=1,500                                                      GRID
       IREC(I) = 0                                                        GRID
    20 CONTINUE                                                           GRID
       JSAVE = 0                                                          GRID
       CALL WRGRID(0)                                                     GRID
       DO 70 I=1,MINDEX                                                   GRID
       CALL RDWETR(IRC)                                                   GRID
       IF(IGROPT .GE. 1.AND. IGROPT .LE. 16000)GO TO 25                   GRID
       WRITE(IOUT,902)                                                    GRID
       GO TO 70                                                           GRID
    25 DO 30 J=1,3                                                        GRID
       SLAT(J) = LAT(J)                                                   GRID
       SLON(J) = LON(J)                                                   GRID
    30 CONTINUE                                                           GRID
       RLAT = SLAT(1) + SLAT(2)/60. + SLAT(3)/3600.                       GRID
       RLON = SLON(1) + SLON(2)/60. + SLON(3)/3600.                       GRID
       IF(LAT(4) .EQ. 1HS)RLAT = -RLAT                                    GRID
       IF(LON(4) .EQ. 1HW)RLON = 360. -RLON                               GRID
       RLAT = DEGTOR*RLAT                                                 GRID
       RLON = DEGTOR*RLON                                                 GRID
       RLATP = ABS(RLAT)                                                  GRID
       R = 124.817404012*TAN((PI - 2.*RLATP)/4.)                          GRID
       A = RLON -PI/18.                                                   GRID
       FI = IO + R*COS(A)                                                 GRID
       FJ = JO - R*SIN(A)                                                 GRID
       IF(RLAT .LT. 0.)FI = FI + 2.*IO                                    GRID
       IGRID = FI  + .5                                                   GRID
       JGRID = FJ  + .5                                                   GRID
       IF(JSAVE .EQ. 0 .OR. JGRID .EQ. JSAVE)GO TO 40                     GRID
       CALL WRGRID(1)                                                     GRID
       CALL RDGRID                                                        GRID
    40 JSAVE = JGRID                                                      GRID
       IF(IREC(IGRID).EQ. 0)GO TO 50                                      GRID
       WRITE(IOUT,903) IGRID,JGRID                                        GRID
    50 IREC(IGRID) = IGROPT                                               GRID
    70 CONTINUE                                                           GRID
       CALL WRGRID(-1)                                                    GRID
       GO TO 5                                                            GRID
   900 FORMAT(I5)                                                         GRID
   901 FORMAT(1H0,61H *** MINDEX IS NOT BETWEEN 0 AND 16000. PROGRAM IS TGRID
      1ERMINATED)                                                         GRID
```

```
  902 FORMAT(1H0,82H *** INDEX NUMBER READ FROM WEATAP IS NOT BETWEEN 1 GRID
     1AND 16000. THE DATA IS IGNORED)                                   GRID
  903 FORMAT(1H0,25H *** THE GRID PAIR IGRID=,I5,7H JGRID=,I5,37H HAS ANGRID
     1 INDEX NUMBER ALREADY ASSIGNED)                                   GRID
      END                                                               GRID
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
● ELT RDGRID,1,760401, 68091     , 1

              SUBROUTINE RDGRID                                            RDGRID
      C       THIS ROUTINE READS 1 RECORD FROM DIRECT ACCESS FILE IGRDTP   RDGRID
              COMMON /FLAG/                                                FLAG
             1INP,IOUT,IKTAPE,IGRDTP                                       FLAG
              COMMON /DIRAC/                                               DIRAC
             1ICDCIN(252),IREC(500),IGRID,JGRID,JSAVE,MAXJ                 DEBUG1
              COMMON /WEATDT/                                              WEATDT
             1IGRDPT,LAT(4),LON(4),IA(500),IB(50 ),IC(30)
              JG = JGRID + 1                                               DEBUG1
              READ(IGRDTP'JG,ERR=10)(IREC(I),I=1,500)
              RETURN                                                       RDGRID
           10 WRITE(IOUT,901)
              STOP
          901 FORMAT(1H0,91H *** AN IRRECOVERABLE I/O ERROR HAS OCCURRED ON READ
             1ING A RECORD FROM THE INDEX MATRIX FILE /10X,31H THE JOB IS BEING
             2ABANDONED ***)
              END                                                          RDGRID
```

```
* ELT RDWETR,1,760401, 68093   , 1
                                                                            RDWETR
      SUBROUTINE RDWETR(IRC)                                                RDWETR
C     THIS ROUTINE READS ONE PHYSICAL RECORD FROM NASA WEATHER TAPE         RDWETR
C                NOT VALID FO UNIVAC                                        DTRAC
      COMMON /DTRAC/                                                        DEBUG1
     1ICDCIN(252),IREC(500),IGRID,JGRID,JSAVE,MAXJ                          FLAG
      COMMON /FLAG/                                                         FLAG
     1INP,IOUT,IWTAPE,IGRDIP                                                WEATDT
      COMMON /WEATDT/
     1IGRDPT,LAT(4),LON(4),IA(500),IB(50 ),IC(30)                           RDWETR
      DATA IC1/30/,IC2/50/                                                  RDWETR
      IF(IRC.NE. 1)GO TO 10
      CALL NTRAN(IWTAPE,2,500,IA,NSTAT,22)
    5 IF(NSTAT+1)500,5,10                                                   RDWETR
   10 I =(IRC-1)*IC2+ 1
      DO 20 IG=1,IC2
      IB(IG) = IA(I+IG-1)                                                   RDWETR
   20 CONTINUE
   50 DECODE(     900,IB(1))IGRDPT,(LAT(I),I=1,4),(LON(I),I=1,4)
      IRC = IRC + 1                                                         RDWETR
      IF(IRC .GT. 10)IRC = 1                                                RDWETR
      RETURN                                                                RDWETR
  500 WRITE(IOUT,901)                                                       RDWETR
      GO TO 515                                                             RDWETR
  515 STOP                                                                  RDWETR
  900 FORMAT(I5,13X,3I2,A1,1X,I3,2I2,A1)                                    RDWETR
  901 FORMAT(1H0,52H*** END OF FILE OCCURRED ON WEATHER TAPE PREMATURELYRDWETR
     1)                                                                     RDWETR
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE WRGRID(IOP)                                            WRGRID
C     THIS ROUTINE CREATES AND/OR WRITES RECORDS ON THE INDEX MATRIX    WRGRID
C     FTE = IGRDTP,                                                     WRGRID
      COMMON /WEATDT/                                                   WEATDT
     1IGRDPT,LAT(4),LON(4),IA(500),IB(50 ),IC(30)
      COMMON /DTRAC/                                                    DTRAC
     1ICDFIN(252),IREC(500),IGRID,JGRID,JSAVE,MAXJ                      DFBUG1
      COMMON /FLAG/                                                     FLAG
     1INP,IOUT,IWTAPE,IGRDTP                                            FLAG
      IF(IOP)30,10,40                                                   WRGRID
   10 DEFINE FILE 4(42,500,U,IDUM)
      CALL NTRAN(IWTAPE,10,22)
      MAX =MAXJ - 1                                                     DFBUG1
      DO 20 I=1,MAX                                                     DFBUG1
      K = I
      FIND(IGRDTP'K)
      WRITE(IGRDTP'K,ERR=200)(IREC(L),L=1,500)
   20 CONTINUE                                                          WRGRID
      IREC(1) = 4HINDE                                                  DFBUG1
      IREC(2) = 4HXMAT                                                  DFBUG1
      FIND(IGRDTP'1)
      WRITE(IGRDTP'1,ERR=200)(IREC(I),I=1,500)
      IREC(1) = 0                                                       DFBUG1
      IREC(2) = 0                                                       DFBUG1
      GO TO 60                                                          WRGRID
   30 JG = JGRID + 1                                                    DFBUG1
      FIND(IGRDTP'JG)
      WRITE(IGRDTP'JG,ERR=200)(IREC(I),I=1,500)
C     CALL CLOSE(IGRDTP,0)
      CALL NTRAN(IWTAPE,11,22)
      GO TO 60                                                          WRGRID
   40 JS = JSAVE + 1                                                    DFBUG1
      FIND(IGRDTP'JS)
      WRITE(IGRDTP'JS,ERR=200)(IREC(I),I=1,500)
   60 RETURN                                                            WRGRID
  200 WRITE(IOUT,901)
      STOP
  901 FORMAT(1H0,91H *** AN IRRECOVERABLE I/O ERROR HAS OCCURRED ON READ
     1ING A RECORD FROM THE INDEX MATRIX FILE /10X,31H THE JOB IS BEING
     2ABANDONED ***)
      END                                                               WRGRID
```

7. TOC

LUMP BOOK III

Table of Contents

PART I

PROBLEM DESCRIPTION FOR LUMP PROGRAM

Problem Description for LUMP Program

1.0 SCOPE

1.1 PROGRAM CAPABILITIES

This program will take substrata historical data cards and sample segment ID, crop window, crop window error and statistical data cards, check the various fields for valid input parameters and write these cards as separate intermediate files on disk. It then reads these intermediate disk files and generates the Segment ID file, crop window file and the substrata historical file.

1.2 METHOD OF PROGRAM DEVELOPMENT

A main driver will do the organizing and will call routines FILGEN to generate the disk files and LUMP to generate the SEGMENT ID FILE, CROP WINDOW FILE and the SUBSTRATA HISTORICAL FILE. Existing utility routines will be used for angle conversion, data conversion and random number generation. Other existing routines will be utilized where deemed necessary.

1.3 OPERATIONAL ASSUMPTIONS

- Only 1 case is run at a time.
- Data Input Cards for the SAMPLE SEGMENTS are sorted by country, region, zone, strata, substrata, segment.
- Data Input Cards for the SUBSTRATA and CROP CALENDAR data are sorted by country, region, zone, strata, substrata.
- Program is a one pass process in which cards are checked, written to disk if no fatal errors, and then LUMP is executed.
- Maximum of 2000 training segments total.
- The input cards for SUBSTRATA HISTORICAL data, SUBSTRATA STATISTICAL data, CROP CALENDAR data, CROP CALENDAR ERROR data, and SAMPLE SEGMENT data are stored on separate input files.
- The United States must be assigned the symbol 'USA ' and Canada must be assigned the symbol 'CAND'.

2.0 INPUT

The only input media to this program is punched cards or card images on disk or tape files. They consist of a control card which includes the case number, random number seed, list option, training segment flag and unit numbers for the five possible types of input files. The other cards are the Substrata Historical Data, Substrata Statistical Data, Sample Segment ID data, the Sample Segment Crop Window data, and Crop Calendar Error data. (These last five sets of data may be card images on a disk or tape file.)

2.1 LIST OF INPUT QUANTITIES

See input data description sheets on Pages 141-143.

2.2 CARD FORMATS

See card format sheets on Pages 144 and 145.

2.3 DECK SETUP

The order of the input i.

1. Control Cards
2. Substrata Historical and Statistical Data Cards
3. Crop Window data cards and Crop Window Error data.
4. Sample segment ID cards.

2.4 RULES FOR ENTERING DATA ON CARDS

2.4.1 General

- Integers must be right justified.
- F-format or D-format numbers must have the decimal point present.
- There is a Crop Window card necessary for spring and winter. If either spring or winter are not required, only one card need be input.

2.4.2 Rules for Specific Fields

See Pages 4-6.

INPUT DATA DESCRIPTION

NAME	SYMBOL	DIMENSION	NOMINAL VALUE	RANGE	UNITS	DESCRIPTION
ICASE		1		0-9999		Identification case number for sample segment data (I5)
ISEG		1		0-9999		Segment number (I4)
ICTRY		1				Four character country name (i.e., USA, USSR)
IREG		1		0-999		Region number (I3)
IZONE		1		0-999		Zone number (I3)
ISTRAT		1		0-9999		Strata number (I4)
ISUB		1		0-9999		Substrata number (I4)
NAMSUB		2				Substrata name (2A4)
SUBAR				0-999999999	acres or hectares	Substrata total area (USA and Canada acres, USSR and others-hectares) (floating point, F10.0)
HISTPW	$\widetilde{PW}$	1		0-100		% wheat in substrata (floating point, F7.3)
ISW		1		S, W		Spring or winter wheat indicator: S = Spring, W = Winter
DEVTPW	δPW	1		-9.999 to 9.999		Deviation to true proportion wheat (floating point, F6.3)
CV1	CV_1	1		0-9.999		Coefficient of variation for year-to-year change in proportion wheat (floating point, F5.3)
CV2	CV_2			0-9.999		Coefficient of variation for within county variation of proportion wheat (floating point, F5.3)
CV3	CV_3			0-9.999		Coefficient of variation for within county variation of proportion mixed pixels (floating point, F5.3)
DEVTPM	δPM			9.999		Deviation of true proportion mixed pixels (floating point, F5.3)
ITYPE				0-3		Card type (Ii) =0 Substrata Historical Data =1 Substrata Statistical Data =2 Substrata Crop Calendar Data =3 Substrata Crop Calendar Error Data =4 Sample Segment Location Data
LAT		1		$\pm$ 65°	Deg, Min	Latitude with N or S indicating + or - respectively (A1, I2, I3)

NAME	SYMBOL	DIMENSION	NOMINAL VALUE	RANGE	UNITS	DESCRIPTION
.ONG		1		$\pm$180°	Deg, Min	Longitude with E or W indicating + or 1 respectively (A1, I2, I3)
:MOT		8		1-12		Date month, day, year of each of 4 phases for start phase and stop phase 4(1X, 6I2) for winter or spring wheat
:DAT		8		1-31		
[YRT		8		64-99		
?SEED		1		1.0-NNNNNNNNNN		Floating point odd whole number used as seed to random number generator (D12.0)
[LIST		1		0.1		List option to list all input data cards or only those in error (I2) =0 Only those in error =1 List all input data cards
[SHD		1				Unit number of Substrata Historical Data (I2)
ISSD		1				Unit number for Substrata Statistical Data (I2)
ISID		1				Unit number for Segment ID Data (I2)
ISCW		1				Unit number for Segment Crop Window Data (I2)
ICWE						Unit number for Crop Calendar Error File (I2) - Set to 38 (8)
ITSFG				0, 1		Training Segment Flag (I2) =0 If all segments are training segments =1 If the training segment list is specified, via segment location data cards.
NAGR		1		1-9999		Number of agricultural segments in a substrata
NAL		1		1-9999		Number of allocated segments in a substrata
EV4		1		0-9.999		Multi-yr Std. deviation of historical wheat area
IGRP		1		1-3		Group no assignment for a substrata
DELTWE	δ_1-δ_4	4		$\pm$1.0		Delta error for winter in predicting the bio window dates, expressed at fraction of the window interval.
IPLNTE	δ_0	1		$\pm$99	Days	Error in the mean planting date of the substrata relative to the true mean value.

INPUT DATA DESCRIPTION (Cont.)

NAME	SYMBOL	DIMENSION	NOMINAL VALUE	RANGE	UNITS	DESCRIPTION
SEGSD	σ	1		0-99	Days	Within substrata variation for winter in crop calendar due to change in latitude, altitude, planting date, etc.
TRIND	-	1	0	0-1		0- training segment, 1- ordinary segment
TLIST	-	6		0-9999		List of segment numbers of training segments associated with the ordinary segment. These segments are to be listed in order of decreasing priority. Highest priority is first.
SW1	-	1		S or W		Spring or winter wheat indicator from
SW2	-	1		S or W		Crop calendar data
ELWE1	$\delta_1-\delta_4$	4		±1.0		Delta error for spring - same as DELTWE
PLNT1	δ_0	1		±99	Days	Same as IPLNTE only for spring
SGSD1	σ	1		0-99	Days	Same as for ISEGSD only for spring

CONTROL CARD

CC 1		9	21	26		31		36		41		46		51		56
Case No.	X	Random No. Seed	X	List Option	X	Training Seg Flag	X	Unit for Subst Hist	X	Unit for Subst Stat	X	Unit for Seg ID	X	Unit for Crop Calen.	X	Unit for Crop Error
I5		D12.0		I2		I2		I2		I2		I2		I2		I2

SUBSTRATA HISTORICAL DATA

CC	Field	Format
6	Country	A4
11	Reg	I3
15	Zone	I3
19	Strata	I4
24	Sub-Strata	I4
28	Sub-Strata Name	A8
36	Sub-Strata Total Area	F10.0
46	Hist. PW	F7.3
53	N_{AGR}	I4
57	N_{AL}	I4
61	Group	I1
80	0	

SUBSTRATA STATISTICAL DATA

CC	Field	Format
6	Country	A4
11	Reg	I3
15	Zone	I3
19	Strata	I4
24	Sub-Strata	I4
28	Deviation True PW	F6.3
34	Coeff. True PW	F5.3
39	Coeff. True PW	F5.3
44	Coeff. True PW	F5.3
49	Ratio True PM	F5.3
54	Multi-Yr Std Dev. WA	F5.3
80	1	

SUBSTRATA CROP CALENDAR DATA

CC	Field	Format
6	Country	A4
11	Reg	I3
15	Zone	I3
19	Strata	I4
24	Sub-strata	I4
	S or W	
29	Crop Phase 1 — Start (Y M D), Stop (Y M D)	6I2
42	Crop Phase 2 — Start (Y M D), Stop (Y M D)	6I2
55	Crop Phase 3 — Start (Y M D), Stop (Y M D)	6I2
68	Crop Phase 4 — Start (Y M D), Stop (Y M D)	6I2
80	2	

1 Card For Spring and 1 For Winter

CROP CALENDAR ERROR DATA

CC	Field	Format
6	Country	A4
11	Reg.	I3
15	Zone	I3
19	Strata	I4
24	Sub-strata	I4
28	BioWin 1 Error Winter δ_1	4F5.3
33	BioWin 2 Error Winter δ_2	
38	BioWin 3 Error Winter δ_3	
43	BioWin 4 Error Winter δ_4	
48	Mean Plt. Date Error	I3
51	Crop Cal. STD Dev.	I2
53	BioWin 1 Error Spring δ_1	4F5.3
58	BioWin 2 Error Spring δ_2	
63	BioWin 3 Error Spring δ_3	
68	BioWin 4 Error Spring δ_4	
73	Mean Plt. Date Error	I3
76	Crop Cal. STD Dev.	I2
80	3	

SAMPLE SEG LOCATION DATA

CC	Field	Format
1	-nt	
6	Country	A4
11	Reg	I3
15	Zone	I3
19	Strata	I4
24	Sub-strata	I4
29	Latitude — N/S, Deg * Min	A1I2 A1I2
36	Longitude — E/W, Deg * Min	A1I3 A1I2
44	S or W	A1
46	Tr. Index	I1
48	Training Segment Priority List — Prior. 1, Prior. 2, Prior. 3, Prior. 4, Prior. 5, Prior. 6	6I4
80	4	

2.5 FILES

The only interface input file is INDMAT-index matrix location file. This file is used to get the index number associated with a segment latitude and longitude. See Section 2.4 of the Users Manual for format and contents.

3.0 PROCESSING

3.1 OVERVIEW

This program is arranged as a driver, an input processor, the main program (LUMP) and various utilities to accomplish their function: when called. The driver reads one data card and then calls the input processor which in turn performs checks on the various input fields as shown in Section 5.2. If no errors, the card image is written to an intermediate file and on option to the output file. If errors exist, the card is written to the output file along with the specified messages. The program will continue reading all the data cards and checking for errors but will terminate writing the intermediate card image file if the errors are fatal. If the errors are non-fatal, then the cards in error will not be written to the intermediate file and processing will continue when all cards have been checked. Additional runs may be made using this same data on the saved files, thus eliminating duplicate processing. After the intermediate files have been written, control is given to the main processor LUMP. Here one record of Substrata data (including a Crop Calendar Record) and one record of Segment ID data is read. The substratum number is checked to see if the segment belongs to this particular substratum. If not, the Substrata data is processed and written to the Substrata Historical file. Then another record of substratum data is read and processing continues. If the segment does belong to the substratum, the index number corresponding to the segment latitude and longitude is computed. The segment is then written to the Segment ID file, and another segment is read in and processed in the same manner until all segments for a particular substratum have been processed. A record is then written to the Crop Window File and the Substrata Historical File and new Substrata and Segment ID cards are read in and processed in the same manner until all cards have been read.

3.2 PROGRAM FLOW

Routine LUMPDR is the main driver which calls the input processor or the subroutine LUMP. Subroutine LUMP is the main body and performs all the needed calculations to generate the desired output files. An overview block diagram is presented on Page 151.

3.3 PROCEDURES AND EQUATIONS

3.3.1 Group No. Adjustment on Substrata Input Group No.

1. If a Substrata is typed a Group III a test is made to make sure there is no sample segment in this substrata. If there is, a message is printed and this situation is considered a fatal error.

2. If a substrata is typed a Group I, a test is made to make sure there is at least 1 sample segment in the substrata, if not a message is printed and the segment is retyped as a Group III.

3. If a substrata is typed a Group II, a test is made to see that there is at least one sample segment in one of the Group II substrata's within a governing strata.

3.3.2 Compute True Proportion of Wheat Using Beta Distribution.

1. Compute XBAR = HSTPW(1 + DEVTPW)/100.0
2. Compute SIGMA = HSTPW * CV1/100.00
3. Compute XI from subroutine BETAD given RSEED utilizing the incomplete beta function as computed via TRW canned routines. See subroutine BETAD writeup for details.
4. Compute TPWSS = XI * 100
5. If the subroutine BETAD can not find an answer for XI then TPWSS is SET = HSTPW.

HSTPW	-	Pseudo proportion of wheat
DEVTPW	-	Input deviation to true proportion of wheat
CV1	-	Input coefficient of variation
TPWSS	-	True proportion of wheat
RSEED	-	Random number seed
XBAR	-	Average proportion of wheat value
SIGMA	-	Standard deviation of proportion of wheat
XI	-	Random value for true proportion of wheat

3.3.3 Ensure Compatible Priority Training Segment Lists for Ordinary Segments.

A check is made to make certain that for each ordinary segment a valid list of priority training segments is given. This test is made only if the ITSFG flag is set for both ordinary and training input.

3.3.4 Compute predicted bio window start and stop dates and associated errors -t' for winter and spring:

$$t'_{o_1} = t_{o_1} + \delta_0$$

$$t'_{f_1} = t_{f_1} + \delta_1(t_{f_1} - t_{o_1}) + \delta_0$$

$$t'_{o_2} = t_{o_2} + \delta_1(t_{f_1} - t_{o_1}) + \delta_0$$

$$t'_{f_2} = t_{f_2} + \delta_1(t_{f_1} - t_{o_1}) + \delta_2(t_{f_2} - t_{o_2}) + \delta_0$$

$$t'_{o_3} = t_{o_3} + \delta_1(t_{f_1} - t_{o_1}) + \delta_2(t_{f_2} - t_{o_2}) + \delta_0$$

$$t'_{f_3} = t_{f_3} + \delta_1(t_{f_1} - t_{o_1}) + \delta_2(t_{f_2} - t_{o_2}) + \delta_3(t_{f_3} - t_{o_3}) + \delta_0$$

$$t'_{o_4} = t_{o_4} + \delta_1(t_{f_1} - t_{o_1}) + \delta_2(t_{f_2} - t_{o_2}) + \delta_3(t_{f_3} - t_{o_3})$$
$$+\delta_4(t_{f_4} - t_{o_4}) + \delta_0$$

$$t'_{f_4} = t_{f_4} + \delta_1(t_{f_1} - t_{o}) + \delta_2(t_{f_2} - t_{o_2}) + \delta_3(t_{f_3} - t_{o_3})$$
$$+ \delta_4(t_{f_4} - t_{o_4}) + \delta_0$$

Error Terms are:

$1 - \delta_0$	$3 - t'_{f_2} - t_{f_2}$	$5 - t'_{f_4} - t_{f_4}$
$2 - t'_{f_1} - t_{f_1}$	$4 - t'_{f_3} - t_{f_3}$	

$t_{o_1}-t_{f_4}$ - Reference crop calendar input start-stop dates

σ - Input planting date error

$\delta_1-\delta_4$ - Input error in predicting acquisition windows

3.3.5 Compute Weather Index No.

The program computes the grid I, J from the latitude and longitude of each segment and then pulls out the index number, from the input pattern matrix file INDMAT.

The equations and procedures are defined in the weather index generation document written by G. Rasmussen. The index pattern matrix will be generated by the utility 'GRID' from the NASA weather tape.

3.3.6 Compute PSEUDO proportion of wheat.

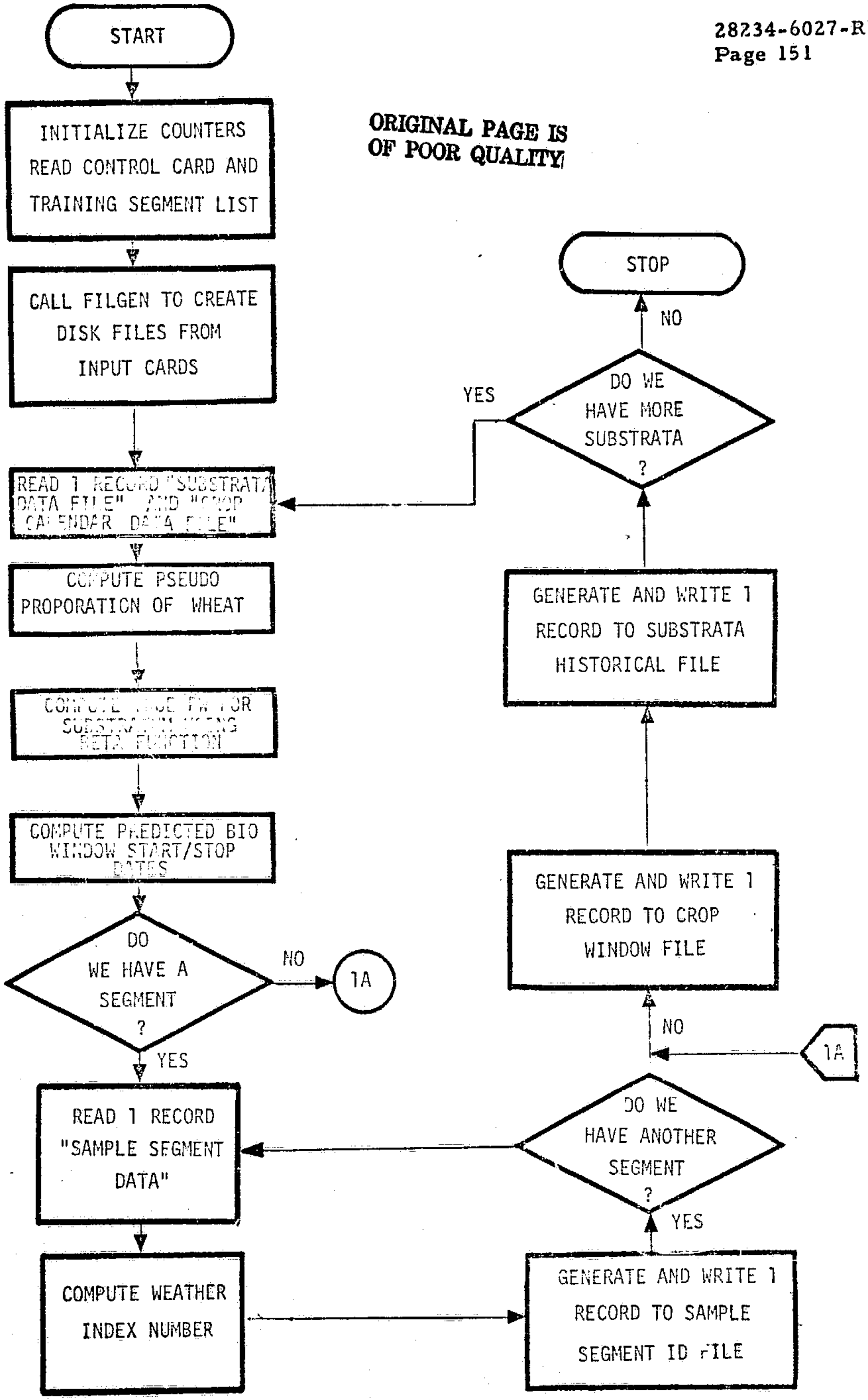
$$RK = \frac{AREA}{(NAL)*10289.712}$$

$$HSTPW = \frac{HISTPW*AREA}{NAGR*.RK*10289.712}$$

where:

AREA - Substrata land area in hectares
NAL - No of allocated segments in a substrata
HISTPW - Historical proportion of wheat in a substrata
NAGR - No of agricultural segments in a substrata

ORIGINAL PAGE IS
OF POOR QUALITY



3.4 Block Diagram and Logic Flow

4.0 OUTPUT

4.1 GENERAL

There are seven media for output. These are:

1. Print file
2. Sample Segment Intermediate File
3. Substrata Data Intermediate File
4. Crop Calendar Intermediate File
5. Segment ID File
6. Crop Window File
7. Substrata Historical File

The second and third are intermediate files and are images of the card input. The last three are the desired resultant of LUMP processing and are to be used later by other modules.

4.2 PRINTED OUTPUT

4.2.1 Standard Printout

The standard printout generated by the LUMP program consists of the following:

1. Listing of the quantities from the "control card"
2. Status recording
   - A count of each type of card read
   - Number of records written to each file
   - A count of input errors
   - A status flag indicating if job ran successfully
   - Last random number seed used

4.2.2 Optional Printout

1. If ILIST $\geq$ 1 on the "control card", then the card images from all six input files (INPUT, Substrata Historical Data, Substrata Statistical Data, Sample Segment ID Data, Crop Window Data, and Crop Window Error Data are printed out.

2. If ILIST = 2, then the contents of the output files (segment ID file, Crop Window File, and Substrata Historical File) are printed out as each record is written (the records for the different files are intermixed on the printed output) although three separate files are actually generated.

4.3 DATA FILES

There are 3 output files: These are the Segment ID file, Substrata Historical file and Crop Calendar files. See Section 2.4 of the Users Manual for format and content.

5.0 ERROR PROCESSING

5.1 GENERAL

The program will attempt to find as many sources of error during input card processing as possible. The program will continue checking for further input errors upon detecting any input error. Processing will continue if possible if recovery from errors may be overcome.

5.2 INPUT DATA ERRORS

- Compatibility checks (Do the country, region zone, strata, and substrata agree between the substrata statistical data and between the sample segment ID data and the crop window data?)
- Checks for non-overlapping windows
- Sequence checking of the card data
- Input/output data limit checks
- Training priority segment NNNN is either not a training segment or nonexistant-fatal.
- A Group 1 substrata does not have at least 1 sample segment. Nonfatal
- A strata with at least 1 group 2 substrata does not have a sample segment in any group 2 substrata . Nonfatal
- A group 3 substrata has 1 or more sample segments . Fatal
- Region xxx zone xxx strata xxxx substrata xxxx beta distribution function could not converge on a solution
- Region xxx zone xxx strata xxxx substrata xxxx beta function reset ----- value since not between 0 and 1. Either ------ or ------ have bad values
- Region xxx zone xxx strata xxxx substrata xxxx segment xxxx. No crop calendar data available for segment intermediate file record, record skipped
- Region xxx zone xxx strata xxxx substrata xxxx segment xxxx. No weather grid index no. could be found. The segment record is skipped.

Input Data Limit Checks

Parameters	Range
Substrata Area	> 0
Historical PW	0 to 10^2
Deviation of True	-9.999 to 9.999
Yr-to-Yr CV of True PW	0 to 9.999
Within County CV of True PW	0 to 9.999
Within County CV of True PM	0 to 9.999
Multi Year CV of Standard Dev	0 to 9.999
Ratio of PM to PW	0 to 9.999
Spring/Winter Wheat Indicator	S or W
Segment Latitude	-65° to +65°
Crop Window Dates	
Year	64 - 99
Month	1 - 12
Day	1 - 31
Grid Index	1 - 16000
DELTWE, DELWE1	-1.0, +1.0

5.3 ERROR MESSAGES

Card Field Errors

For any error on a card the card image is printed pr·or to the message:

1. Inconsistent substrata historical and statistical data
2. Inconsistent Crop Window Error Data and Crop Window Data
3. Substrata area must be positive
4. Historical proportion of wheat must be non-negative
5. DEVTPW must be less than 9.999 in magnitude
6. COEFF, if variations must be between 0.0 and 9.999
7. DEVTPM must be between 0.0 and 9.999
8. Latitude must be less than or equal to 65.0 deg.
9. Longitude must be between -180 and 180 deg.
10. Spring/winter wheat indicator must be W or S
11. CV4 must be between 0.0 and 9.999
12. Year must be greater than 64
13. Month must be between 1 and 12
14. Day must be between 1 and 31
15. Overlapping crop window
16. Group no must be equal to 1, 2 or 3
17. Delta errors in bio window prediction dates are not between -1 and +1
18. Segment specified as ordinary must have a training segment priority list.
19. Neither NAGR or NAL can be zero.

PART II

COMMON BLOCK DEFINITIONS

COMMON STORAGE ALLOCATION

Name CONST Size 18 words Page 1 of 2

Function Program Constants

Name	Dimension	Format	Description	Symbol	Units
PI	1	F9.7	$\pi = 3.1415926$	π	
DEGTOR	1	F11.9	Degrees to radians conversion factor $= \pi/180$		radians / deg.
AREACF	1	F11.6	Area conversion factor = 4.046873E-3 KM^2/acre		KM^2 / acre
			For U.S.A. and Canada = 0.01 KM^2/hectares otherwise		KM^2 / hectare
SLATMX	1	F4.1	Minimum allowable latitude (65.0)		deg.
ENDATA	1	A4	End-of-data indicator (=4HZZZZ)		
ZERO	1	I2	Binary zero (0)		
INP	1	I2	Input file number (5)		
IOUT	1	I2	Output file number (6)		
INTSUB	1	I2	Substrata Intermediate File (30)		
INTSEG	1	I2	Segment Intermediate File (31)		
ISIDF	1	I2	Segment ID File number (1)		
ICWF	1	I2	Crop window file number (2)		
ISHF	1	I2	Substrata historical file number (3)		
LRSIDF	1	I2	Record length for the Segment ID File (13)		
LRCWF	1	I2	Record length for the Crop Window File (21)		
LRSHF	1	I2	Record length for the Substrata historical file (160)		
IWIN	1	I2	Number of crop windows (4)		
MXSEG	1	I2	Maximum number of segments in any substrata (150)		

COMMON STORAGE ALLOCATION

Name CONST Size Page 2 of 2

Function Program Constraints

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
ICALIN	1	I	Unit number of intermediate crop calender file		
ICWE	1	I	Unit number of crop calender error data		
ISW1	1	I	Spring or winter wheat input flag from crop calender cards		
ISW2	1	I	Spring or winter wheat input flag from crop calender cards		
IGRDTP	1	I	Unit number of index matrix file		

COMMON STORAGE ALLOCATION

Name FLAGS Size 182 Page 1 of 4

Function Flags and Indexes

Name	Dimension	Format	Description	Symbol	Units
IERR	1	I2	Number of input errors		
IPRNT	1	I2	Print flag used in ERRMES ≠ 0 if the current card image has already		
			been printed = 0 otherwise		
GOOD1	1	I2	Error flag: ≠ 0 if the current substrata historical data		
			card or the current sample segment ID data card contains errors;		
			= 0 otherwise		
GOOD2	1	I2	Error flag: ≠ 0 if the current substrata statistical data		
			card or the current sample segment crop window data card contains errors;		
			= 0 otherwise		
NSHC	1	I2	Number of substrata historical data cards		
NSSC	1	I2	Number of substrata statistical data cards		
NSEGC	1	I2	Number of sample segment data cards		
NSUBR	1	I2	Number of records written on the substrata intermediate file		
NSEGR	1	I2	Number of records written on the segment intermediate file		
NRSIDF	1	I2	Number of records written on the segment ID File		
NRCWF	1	I2	Number of records written on the Crop Window File		
NRSHF	1	I2	Number of records written on the Substrata Historical File		

COMMON STORAGE ALLOCATION

Name FLAGS Size Page 2 of 4

Function

Name	Dimension	Format	Description	Symbol	Units
ISSC	1	I2	Substrata Statistical Data File read flag: ≠ 0 if the statistical data file was read		
			after reading the historical data file; = 0 otherwise		
ISTAT	1	I2	Substrata Statistical Data read flag: = 0 if for each substrata Historical Data		
			card, a statistical data card is to be read;		
			= 1 if the current statistical data card is to be used for all substrata within the		
			current strata; = 2 if the current statistical data card is to		
			be used for all strata within the current zone;		
			= 3 if the current statistical data card is to be used for sll zones within the		
			current region; = 4 if the current statistical data card is to		
			be used for all regions		
IREGL	1	I2	Previous region number		
IZONEL	1	I2	Previous zone number		
ISTRL	1	I2	Previous stratum number		
ISUBL	1	I2	Previous substratum number		
ISEGL	1	I2	Previous segment number		
NSEG	1	I2	Number of segments within the current substratum		
IMXSEG	1	I2	Maximum number of segments within any substrata		
NTLIST	1	I2	Number of segments in the training segment ID list		

COMMON STORAGE ALLOCATION

Name FLAGS Size Page 3 of 4

Function

Name	Dimension	Format	Description	Symbol	Units
READSF	1	I2	Segment intermediate file read flag:		
			≠ 0 to skip reading of the segment intermediate file until a new substratum is read on the Substrata Intermediate File;		
			= 0 to read the segment intermediate file		
FATAL	1	I2	≠ 0 if a fatal error is detected in the input data		
IDAYS	8	I2	Array of crop window start/stop dates in Zulu date format:		
			IDAYS(2I-1) = start date IDAYS(2I) = stop date		
			for the I-th crop window for winter		
IDSEG	150	I2	List of segment ID numbers for a given substratum		

COMMON STORAGE ALLOCATION

Name FLAGS Size ______ Page 4 of 4

Function ______

Name	Dimension	Format	Description	Symbol	Units
IMOT	8		Temporary storage for crop calender stop/start dates Card 1 - Month		
IDAT	8		Card 1 - Day		
IYRT	8		Card 1 - Year		
IYR1	8		Card 2 - Year		
IMO1	8		Card 2 - Month		
IDA1	8		Card 2 - Day		

COMMON STORAGE ALLOCATION

Name INPUT Size 1071 Page 1 of 6

Function Input Quantities

Name	Dimension	Format	Description	Symbol	Units
RSEED	1	F10.0	Random number seed		
SUBAR	1	F10.0	Substratum Total Area (acres for USA and Canada; hectares otherwise)		acres or hectares
HISTPW	1	F7.0	Historical proportion of wheat (% wheat) converted to correct units for use in subroutine LUMP	$\widetilde{PW}$	
DEVTPW	1	F6.3	Deviation of true proportion of wheat	δPW	
CV	3	F5.3	Coefficient of variation for year-to-year change in proportion of wheat	CV_1	
			Coefficient of variation for within county variation of proportion of wheat	CV_2	
			Coefficient of variation for within county variation of proportion of mixed pixels	CV_3	
DEVTPM	1	F5.3	Deviation of true proportion of mixed pixels	δPM	
QLAT	1	E10.6	Segment latitude		radians
QLONG	1	E10.6	Segment longitude		radians
SLAT	3		Segment latitude:		
		F4.0	SLAT(2) = degrees portion		
		F4.0	SLAT(3) = minutes portion		
SLONG	3		Segment longitude:		
		F4.0	SLONG(2) = degrees portion		
		F4.0	SLONG(3) = minutes portion		

COMMON STORAGE ALLOCATION

Name INPUT Size ______ Page 2 of 6

Function ______

Name	Dimension	Format	Description	Symbol	Units
NAMSUB	2	A4	Substrata name		
ICTRY	1	A4	Name of country on substrata Historical data card or on Sample Segment ID data card		
ICTRY2	1	A4	Name of country on Substrata Statistical Data card or on Sample Segment Crop Window Data card		
ICASE	1	I5	Case number		
ILIST	1	I2	List option = 0 for no printout except errors and statistics = 1 to printout input data = 2 to printout input data and the output files		
ITSFG	1	I2	Training segment flag ≠ 0 if the training segment list is specified = 0 if no training segment list is specified and if all segments are training segments		
ISHD	1	I2	Unit number for Substrata Historical Data		
ISSD	1	I2	Unit number for Substrata Statistical Data		
ISID	1	I2	Unit number for Sample Segment Statistical Data		
ISCW	1	I2	Unit number for Sample Segment Crop Window Data		
ISEG	1	I4	Segment ID (on Segment ID Data Card)		
IREG	1	I3	Region number		
ISZONE	1	I3	Zone number		
ISTRAT	1	I4	Strata Number		

COMMON STORAGE ALLOCATION

Name INPUT Size ______ Page 3 of 6

Function ______

Name	Dimension	Format	Description	Symbol	Units
ISUB	1	I4	Substrata number Note: IREG, IZONE, ISTRAT, and ISUB		
			refer to the Substrata Historical Data or the Segment ID Data; IREG2, IZONE2, ISTRA2		
			and ISUB2 refer to the Substrata statistical data or the segment crop window data.		
ISW	1	A1	Spring/winter wheat indicator: = S for spring wheat		
			= W for winter wheat		
ITYPE	1	I1	Card type = 0 for substrata hist. data		
			= 1 for substrata statistical data = 2 for segment ID data		
			= 3 for segment crop window data		
ITYPE2	1	I1	Card type - similar to ITYPE		
			ITYPE is used for the substrata historical data and the segment ID data and ITYPE2 is used		
			for the substrata statistical data and the crop window data		
INDEX	1	I5	Segment index (grid number)		
ISEG2	1	I4	Segment ID (on crop window data card)		
IREG2	1	I3	Region number		
IZONE2	1	I3	Zone number		
ISTRA2	1	I4	Strata number		
ISUB2	1	I4	Substrata number Note: See the note following description of		
			ISUB above		

COMMON STORAGE ALLOCATION

Name INPUT Size ______ Page 4 of 6

Function ______

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
LAT	3	A1	Segment latitude		
		I2	LAT(1) is N or S LAT(2) is the degrees portion		degrees
		I2	LAT(3) is the minutes portion		minutes
LONG	3	A1	Segment longitude		
		I2	LONG(1) is E or W LONG(2) is the degrees portion		degrees
		I2	LONG(3) is the minutes portion		minutes
IYR	8	I2	Year portion of the start/stop dates (≥ 64)		years
IMO	8	I2	month portion of the start/stop dates (1 - 12)		months
IDA	8	I2	day portion of the start/stop dates (1-31)		days
			IYR(2n-1) is the start date of the nth crop window		
			IYR(2n) is the stop date of the nth crop window		

COMMON STORAGE ALLOCATION

Name INPUT Size ______ Page 5 of 6

Function ______

Name	Dimension	Format	Description	Symbol	Units
IGRP	1	I	Substrata group number		
CV4	1	R	Multi year standard deviation		
DELTWE	4	R	Bio window error set for winter	δ_i	
IPLNTE	1	I	Mean planting date error for winter	δ_o	
ISEGSD	1	I	Crop calender standard deviation for winter		
ITRIND	1	I	Training segment indicator		
NAGR	1	I	Number of agricultural segments in a substrata		
NAL	1	I	Number of allocated segments in a substrata		
ITLIST	6	I	List of priority training segments		
NCRCD	1		Number of crop calender cards read		
NCRECD	1		Number of crop calender error cards read		
NCROP	1		Number of records written on crop calender file		
IDAYS1	8		Crop calender start/stop zulu dates for spring		
DELWE1	4		Bio window error set for spring		
IPLNT1	1		Mean planting date error for spring		
ISGSD1	1		Crop calender standard deviation for spring		
IDAS	8		Input crop calender start stop set for spring day set		
IMOS	8		month set		

COMMON STORAGE ALLOCATION

Name INPUT Size Page 6 of 6

Function

Name	Dimension	Format	Description	Symbol	Units
IYRS	8	I	Year set		

COMMON STORAGE ALLOCATION

Name LDATA   Size 16   Page 1 of 1

Function Spring/Winter Wheat Crop Window Dates

Name	Dimension	Format	Description	Symbol	Units
WWDATA	8	I5	Winter wheat crop window start/stop dates (Zulu Dates - no. of days since 1 Jan 1950)		
SWDATA	8	I5	Spring wheat crop window start/stop dates		
			(Zulu Dates) WWDATA(2n-1) is the start date		
			WWDATA(2n) is the stop date for the nth crop window		
			Similarly for SWDATA		

COMMON STORAGE ALLOCATION

Name CALEND Size Page 1 of 1

Function

Name	Dimension	Format	Description	Symbol	Units
IPREDW	8	I	Predicted start and stop date set for winter	t'_{o_i}, t'_{f_i}	
IPREDS	8	I	Predicted start and stop date set for spring	t'_{o_i}, t'_{f_i}	
ICRPEW	8	I	Crop window error set for winter		
ICRPES	8	I	Crop window error set for spring		

COMMON STORAGE ALLOCATION

Name TRATAB Size ____ Page 1 of 1

Function Training Table Data

Name	Dimension	Format	Description	Symbol	Units
ITRAIN	2000	I	List of all training segments specified via sample segment location data		
ITRPRI	2000	I	List of unique segment IDs identified in priority lists for ordinary segments		

COMMON STORAGE ALLOCATION

Name DIRAC Size Page 1 of 1

Function Direct Access Storage Index Matrix File

Name	Dimension	Format	Description	Symbol	Units
ICDIN	252	I	Array for index record pointers for CDC only		
IREC	500	I	Contains 1 physical record from the index matrix file		

CROP CALENDAR

Intermediate Data File - CRPINT

This file contains the substrata crop calendar and its corresponding error data stored in one record. This file is used by subroutine LUMP.

Access Method: Sequential

Status: Temporary

Sort: Country through substrata

Media: Drum

Record Format:

Detail Record

ICTRY - Country ID - A4
IREG - Region no.
IZONE - Zone no.
ISTRAT - Strata no.
ISUB - Substrata
IDAYS - Four stop/start Zulu date pairs for winter
DELTWE - Four values of bio-window errors for winter
IPLNTE - Winter mean planting date error
ISEGSD - Winter crop calendar standard deviation
IDAYS1 - Four stop/start Zulu date pairs for spring
DELWE1 - Four values of bio-window errors for spring
IPLNT1 - Spring mean planting date error
ISGSD1 - Spring crop calendar standard deviation
Total - 33 words

Trailer Record

First word has 'ZZZZ' followed by 32 words of zero

Recommended Blocking Factor: 10

File Size: 105,633 words max.

SUBSTRATA DATA

Intermediate File - SUBINT

This file contains the combined data from the substrata historical and statistical data card sets. It is used by subroutine LUMP.

Access Method: Sequential

Status: Temporary

Sort: Country through substrata

Media: Drum

Record Format:

Detail Record

ICTRY	-	Country ID - A4
IREG	-	Region no.
IZONE	-	Zone number
ISTRAT	-	Strata no.
ISUB	-	Substrata
NAMSUB	-	Two word substrata name - 2A4
AREA	-	Substrata area
HSTPW	-	Historical proportion of wheat converted to proper units
DEVTPW	-	Deviation of true proportion of wheat
CV	-	Three coefficients of variation
DEVTPM	-	Deviation of true proportion of mixed pixels
NAGR	-	No. of agricultural segments
NAL	-	No. of allocated segments
CV4	-	Multi-year standard deviation
IGRP	-	Group no.

Total - 18 words

Trailer Record:

First word has 'ZZZZ' followed by 17 words of 0

Recommended Blocking Factor: 10

File Size: 57,618 words max.

SEGMENT DATA

Intermediate File - SEGINT

This file contains the data from the sample segment cards. It is used by subroutine LUMP.

Access Method: Sequential

Status: Temporary

Sort: Country through segment

Media: Drum

Record Format:

Detail Record:

ICTRY	-	Country ID - A4
ISEG	-	Segment no.
IREG	-	Region no.
IZONE	-	Zone no.
ISTRAT	-	Strata no.
ISUB	-	Substrata no.
QLAT	-	Latitude
QLONG	-	Longitude
ISW	-	Spring/winter flag
ITRIND	-	Training segment flag
ITLIST	-	Six training segment ID's

Total - 16 words

Trailer Record:

First word has 'ZZZZ' followed by 15 words of 0

Recommended Blocking Factor: 10

File Size: 64,000 words max.

PART III

LIST OF SUBROUTINES AND SUBROUTINE CALL STRUCTURE

LIST OF ROUTINES IN LUMP

LUMPDR	Main control routine for the LUMP program.
FILGEN	Reads data from the four input files and generates the Substrata Intermediate File and the Segment Intermediate File.
LUMP	Reads data from the Substrata Intermediate File and from the Segment Intermediate File and generates the Segment ID file, the Crop Window file, and the Substrata Historical file.
ERRMES	Error Message routine. Prints most of the error messages.
LFPA	Calculates the Zulu Date (Number of days from 1 January 1950) from a given calendar date (year, month, day).
RDMIA	Random number generator.
CALGEN	Reads, checks and merges crop calender and crop calender error data. Writes intermediate file.
CALERR	Computes predicted bio window start and stop dates with their associated errors.
BETAD	Compute a random number based on the beta distribution or normal distribution.
IBETAI	Computes incomplete beta function (TRW utility)
ALGAMA	Computes gamma function (TRW utility)
GRID	Controls the calculation of the weather grid index number. given latitude and longitude
GRIDCAL	Computes the best index value for a given grid pair of coord. in the global stereographic index matrix.
RDGRID	Opens or closes IGRDTP file or reads a record from it (the Index Matrix File).

SUBROUTINE CALL STRUCTURE

```
LUMPDR
    RDGRID
    FILGEN
        ERRMES
        CALGEN
            ERRMES
        LFPA
    LUMP
        CALERR
        BETAD
            RDMIA
            IBETAI
                ALGAMA
        GRID
            GRDCAL
                RDGRID
```

PART IV

SUBROUTINE DESCRIPTIONS AND FLOWCHARTS

SUPPLIED UTILITY ROUTINES

Routine Day

Call Day (IYMD, IDAY)

Given IYMD (3) where	IYMD (1)	IS Day No.
	IYMD (2)	IS Month No.
	IYMD (3)	IS Year No.

Compute year day no. in IDAY

Routine PIMOD

Call PIMOD (A)

Convert ±A in radians to an angle $0 - 2\pi$

Routine SOL (Entry ALPHA)

Call ALPHA (IFLAG)

For emphemeris usage as called by hector

computes ALPHAM and ALPHAT and IFLAG = 1

Routine PAGER (Entry Eject)

Call PAGER (NLINES)

Updates line count in NLINE with NLINES

NPAGE = 0 causes page to be restored prior to print.

NPAGE - page no.

HEADER - 80 char. 20A5

ICASE - case no.

KO - 6 print unit

INMAX is max no. of lines allowed

Initially NLINE should be set > LINMAX and NPAGE = 0

SUPPLIED UTILITY ROUTINES
(CONTINUED)

Call EJECT (NLINES)

Causes page to be restored automatically and then prints headers.

Routine CLDAY

Call CLDAY

Given IDAY-DAY no. of the year compute in LMO-the month and in LDA the day no.

Need: IYEAR = 0 - Leap Year, ≠ 0 not Leap Year

Routine KEPLER

Call KEPLER (XM, XECC, XE, ERROR)

Given XM - Mean anomaly, XECC - eccentricity

Compute: E-eccentric anomaly, error = 0 means OK

Routine LFPA

Call LFPA [FLDA, LMO, LYR, ALFGM (can be dummy), DAYS]

Given: FLDA - day of month no., LMO - month no., LYR - year no. compute ALFGM - right ascension and DAYS = Zulu day no.

Routine DEGMOD

Call DEGMOND (RAD, IDEG)

Given: angle rad in radians store the angle in deg., min., sec., in IDEG(1) - (3).

Routine FZULU

Call FZULU (IOATE, IOUT)

Given Zulu date in IDATE, compute year, month and day in IOUT(1) - IOUT(3).

Routine RDMIA

Call RDMIA(FL, U)

Given double precision random no. seed in FL, comoute random no. U (0-1) based on uniform distribution.

Subroutine LUMPDR

Purpose: Main control routine for the LUMP program

Linkage: Entered from the Operating System

Inputs:

The input quantities for LUMPDP are read from the INPUT file and consist of all the information on the "Control" card plus the Training segment ID list.

Outputs:

The output quantities from LUMPDP are the quantities read from the INPUT file and are stored in the labeled COMMON block /INPUT/ for use by other routines. In addition the quantity NTLIST, the number of entries in the Training Segment ID list is computed and stored in the COMMON block /FLAGS/.

Subroutines Called:

FILGEN - reads data from the four input files and generates the Substrata Intermediate File and the Segment Intermediate File.

LUMP - reads data from the two intermediate files and generates the Segment ID File, the Crop Window File, and the Substrata Historical File.

RDGRID - read and check header record on IND mat file

Error Messages:

None

Definitions of Local Variables:

I Index in DO LOOP to read Training Segment ID list.

J Index in DO LOOPs to read Training Segment ID list.

L1 Lower limit of J in the DO LOOP for a given card image.

L2 Upper limit of J in the DO LOOP for a given card image.

Subroutine FILGEN

Purpose: Reads data from three input files and generates the Substrata Intermediate Files and the Segment Intermediate File.

Linkage: CALL FILGEN

There are no arguments in the calling sequence. All input/ourput quantities are transmitted through COMMON storage or disk/tape/drum files.

Input:

Most of the input quantities are specified on three input files

1. Substrata Historical Data
2. Substrata Statistical Data
3. Sample Segment ID Data

In addition, the following quantities are required and are specified through COMMON storage:

ILIST	List option: = 1 to list input data; = 0 otherwise
ISHD	File number for Substrata Historical Data (=0 if the Substrata Intermediate File from a previous run is to be used)
ISSD	File number for Substrata Statistical Data (=0 if the Substrata Intermediate File from a previous run is to be used)
IOUT	File number for the print file on which card images, error messages, and run statistics may be printed
ISID	File number for Sample Segment ID Data (= 0 if the Segment Intermediate File from a previous run is to be used)

ENDATA	End of data indicator (= 4HZZZZ)
DEGTOR	Degrees to radians conversion factor (= π/180)
INTSUB	File number for Substrata Intermediate File
INTSEG	File number for Segment Intermediate File

Output:

The output data generated by FILGEN is written on the Substrata Intermediate File and the Segment Intermediate File. In addition, the following statistical quantities are computed and printed out:

NSHC	Number of Substrata Historical Data cards read
NSSC	Number of Substrata Statistical Data Cards read
NSEGC	Number of Sample Segment ID Data cards read (also the number of Sample Segment Crop Window Data cards)
NSUBR	Number of records written on the Substrata Intermediate File
NSEGR	Number of records written on the Segment Intermediate File
IERR	Number of errors detected on the input data

Subroutines Called:

ERRMES	Error message subroutine
CALGEN	Reads crop calender data and writes intermediate file

Error Messages:

The following error codes may be used. See the writeup of subroutine ERRMES for the complete error message:

3, 4, 5, 6, 7, 8, 9, 10, 11, 16, 18, 19

Definitions of Local Variables:

AREA	Area of substratum in Km^2
DAYS	Zulu date (in floating point)
FLDA	Day portion of calendar date
HSTPW	Historical proportion of wheat (true percent)
I	Index in miscellaneous DO LOOPs.
ISKCW	Flag controlling reading of Sample Segment Crop Window Data: = 0 to read one card image from the Crop Window Data after reading one card image from the Segment ID Data = 1 to skip reading the Crop Window Data temporarily until the Segment ID Data catches up with the Crop Window Data (i.e., until the segment IDs match)
ISKSSC	Flag controlling reading of the Substrata Statistical Data: = 1 to skip reading of the Substrata Statistical Data until the Substrata Historical Data catches up with the Statistical Data (i.e., until the region, zone, strata, and substrata agree unless the region, zone, strata, or substrata from the statistical data are zero)
ICT	Count of number of training segments associated with and ordinary segment specified in a priority list.
ILOC	Current location in ITRAIN array
ILOCI	Current location in ITRPRI array
ISEGPR	Current priority training segment ID used to make sure segment input data has specified valid input priority training segment lists.

Subroutine LUMP

Purpose: Reads data from the Substrata Intermediate File Crop Calender Intermediate File and the Segment Intermediate File and generates the Segment ID file, the Crop Window File, and the Substrata Historical File.

Linkage: CALL LUMP

There are no arguments in the calling sequence. All input/output quantities are transmitted through COMMON storage or disk/tape/drum files.

Input:

Most of the input quantities are specified on the Substrata Intermediate File and the Segment Intermediate File. In addition, the following quantities are required and are specified through COMMON storage:

INTSUB	File number for the Substrata Intermediate File
INTSEG	File number for the Segment Intermediate File
ISHF	File number for the Substrata Historical File
ISIDF	File number for the Segment ID File
ICWF	File number for the Crop Window File
LRSHF	Length of one record of the Substrata Historical File
LRSIDF	Length of one record of the Segment ID File
LRCWF	Length of one record of the Crop Window File
ICASE	Case number
RSEED	Random number seed
ICALIN	File number of crop calender intermediate file

ILIST	List option = 1 to print out input card images (in subroutine FILGEN) = 2 to print out the contents of the three output files = 0 to suppress all printing except for error messages and statistics
ITSFG	Training Segment option = 1 if the Training Segment ID list is used; = 0 if all segments are training segments
IOUT	File number for print file
IWIN	Number of crop windows (4)
MXSEG	Maximum number of segments in any substrata (150)
ENDATA	End-of-data indicator (4HZZZZ)
ZERO	Zero constant

Output:

The output data is written on the following three files:

1. Segment ID File
2. Crop Window File
3. Substrata Historical File

Subroutines Called:

CALERR	Computes predicted bio window start/stop dates
BETAD	Generate random number using incomplete beta distribution
GRID	Computes the weather index for segments latitude and longitude

Error Messages:

1. "THE FOLLOWING SEGMENT DATA WAS DROPPED DUE TO DISAGREEMENT WITH THE SUBSTRATA DATA." followed by a printout of the segment, country, region, zone, stratum, and

substratum from the current record of segment data from the Segment Intermediate file and from the current record of substrata data from the Substrata File. This error message indicates that the indicated set of Segment Data cannot be matched with any substrata on the Substrata Intermediate File. Possibly one file or the other is out of order.

2. "THE FOLLOWING SEGMENT DATA WAS DROPPED BECAUSE IT DID NOT BELONG TO ANY STRATA ON THE SUBSTRATA HISTORICAL FILE" followed by a printout of the segment, country, region, zone, stratum, and substratum from the current record of the Segment Intermediate File. This message indicates that the indicated segment is being dropped because there are no more data records available on the Substrata Intermediate File. Hence there is no way to match up the indicated segment with the proper set of Substrata Data. Again one file or the other may be out of order.

3. "REGION NNN ZONE NNN STRATA NNNN SUBS NNNN BETA FUNCTION COULD NOT CONVERGE ON A SOLUTION." This message means that subroutine BETAD could not on a solution for RSEED.

4. "REGION NNN ZONE NNN STRATA NNNN SUBSTRATA NNNN SEGMENT NNNN NO CROP CALENDAR DATA AVAILABLE FOR SEGMENT INTERMEDIATE FILE RECORD. SEGMENT RECORD SKIPPED." Self explanatory. The ID is for the segment record being skipped.

5. "REGION NNN ZONE NNN STRATA NNNN SUBSTRATA NNNN A GROUP 3 SUBSTRATA HAS 1 OR MORE SAMPLE SEGMENTS." A group 3 substrata must not have any sample segments. This error is job fatal.

6. "REGION NNN ZONE NNN STRATA NNNN SUBSTRATA NNNN A GROUP 1 SUBSTRATA DOES NOT HAVE AT LEAST 1 SAMPLE SEGMENT." A group 1 substrata is supposed to have 1 or more sample segments. The group number is reset to 3.

7. "REGION NNN ZONE NNN STRATA NNNN A STRATA WITH AT LEAST 1 GROUP 2 SUBSTRATA DOES NOT HAVE A SAMPLE SEGMENT IN ANY GROUP 2 SUBSTRATA." A group 2 strata must have at least 1 sample segment.

8. "REGION NNN ZONE NN STRATA NNNN SUBSTRATA NNNN BETA FUNCTION RESET AAAAAA VALUE SINCE NOT BETWEEN 0 AND 1.
EITHER AAAAAA OR AAAAAA HAVE BAD VALUES, NNN.NNN NNN.NNN." Subroutine BETAD can not operate outside the specified limits.

Definitions of Local Variables:

AREA	Area of a substratum in Km^2
CWHDR	Two word array containing the header for the Crop Window File in the format 4HCROP, 4HWIND
I	Index in DO LOOPs
ISEGR	Index for number of records read from the Segment Intermediate File
ISTRTP	Strata type flag: = 1 if there is at least one segment in the current substratum; = 0 otherwise
NFILL	Number of words required to complete the header record or the trailer record of one of the output files

SHHDR	Two word array containing the header for the Substratum Historical File in the format 4HSUB, 4HHIST
SIDHDR	Two word array containing the header for the Segment ID File in the format 4HSEGM, 4HENT
TPWSS	PW_K = Temporary location for * proportion wheat for a substratum

IREGP, IZONP, ISTRP - Previous substrata history intermediate file ID (Reg. - Strata)

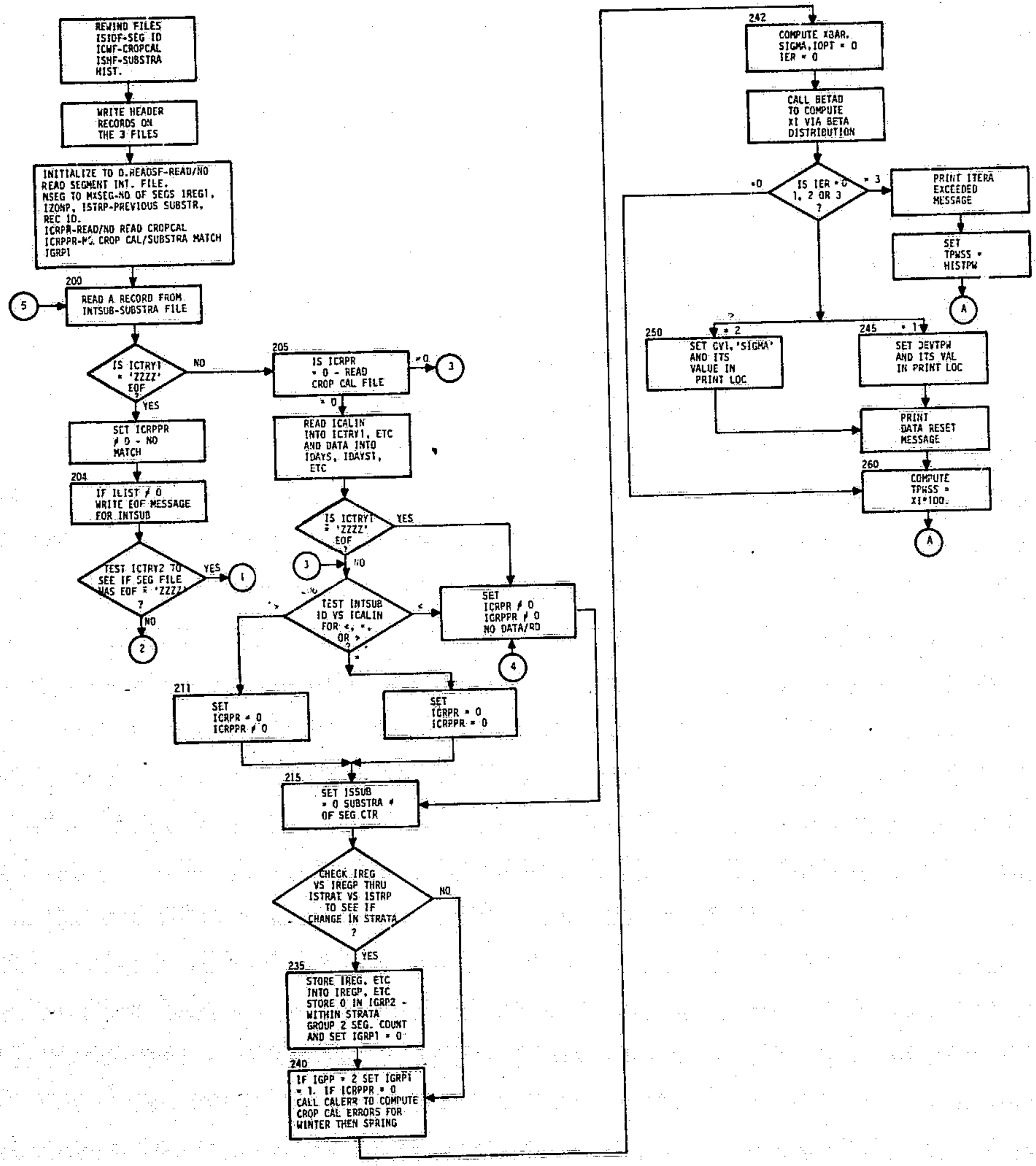
ICRPR	≠ 0 Do not read crop calendar intermediate file = 0 Okay to read
ICRPPR	≠ 0 Crop calendar ID does not match substrata history intermediate file = 0 IDs match
ISSUB	Number of segments in a substrata (used in group processing logic)
IGRP2	Number of Group 2 segments in a strata
ICV1	Constant 'CV1'
IDEVTP	Constant 'DEVTPW'
IHISTP	Constant 'HISTPW'
ISIGMA	Constant 'SIGMA'

ICTRY1, IREG1, IZON1, ISTRA1, ISUB1 - Crop calendar intermediate file ID

IGRP1	Number of Group 1 segments in a strata
ISWH	Flag indicating whether segment data is spring = 1 or winter = 0
XI	Random number output from BETAD
XBAR	Mean value of variable input to BETAD
SIGMA	Standard deviation of variable

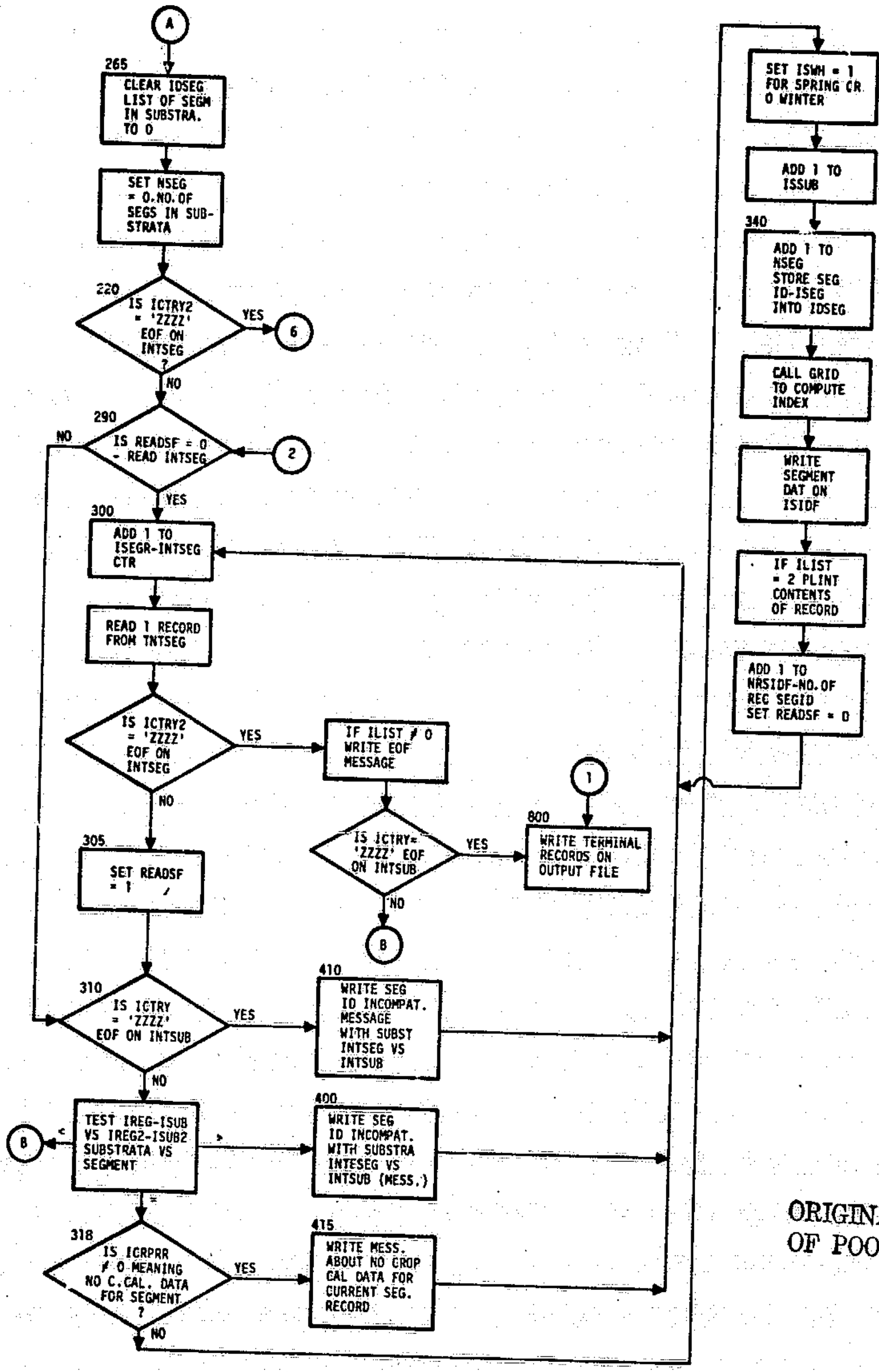
C-3

LUMP FLOWCHART



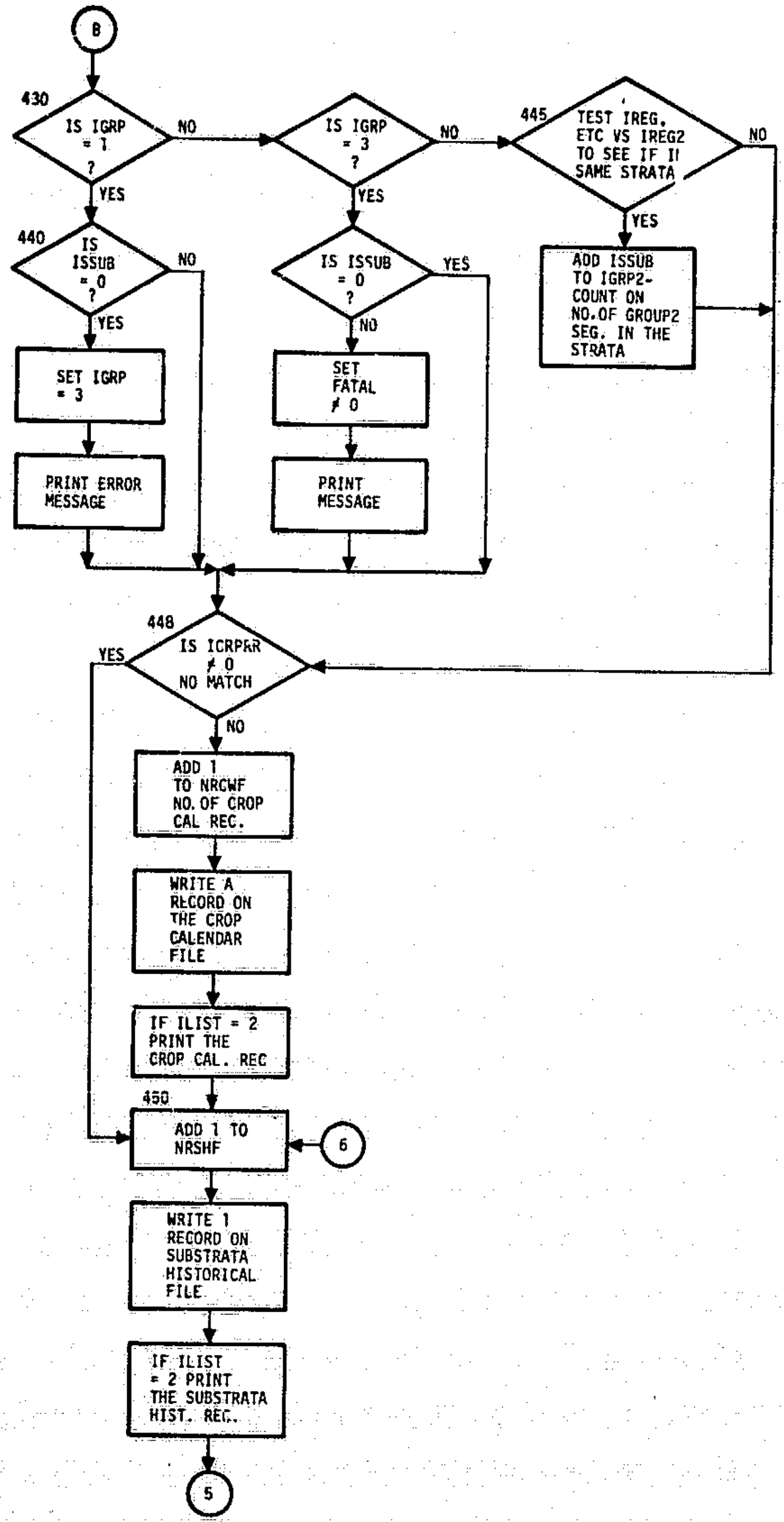
ORIGINAL PAGE IS
OF POOR QUALITY

LUMP (CONTINUED)



ORIGINAL PAGE IS OF POOR QUALITY

LUMP FLOWCHART



ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE ERRMES

Purpose: Error message subroutine: Writes error messages on the print file as directed by flags in the calling sequence.

Linkage: Call ERRMES (ICARD, ICODE)

Input: ICARD - Type of card containing the error.

= 1 for substrata historical data;

= 2 for substrata statistical data;

= 3 for sample segment ID data;

= 4 for substrata crop widow data.

= 5 for crop window error data.

ICODE - Error code (1 - 19)

In addition to the above two quantities specified through the argument list, the following input quantities are also required and are transmitted through labeled common:

IPRNT - Card image print flag:

= 0 if the current pair of card images have not been printed;

≠ 0 if they have been printed.

IERR - Previous number of input errors.

ICTRY IREG IZONE ISTRAT ISUB ISEG	The country, region, zone, stratum, substratum and segment from either the Substrata Historical Data, the Sample Segment Data or substrata crop window data.
ICTRY2 IREG2 IZONE2 ISTRA2 ISUB2 ISEG2	The country, region, zone, stratum, substratum and segment from either the Substrata Statistical Data or the Crop Window Error Data.

NAMSUB SUBAR HISTPW ITYPE NAGR NAL IGRP	The substrata name, area, Historical PW, etc., and card type from the Substrata Historical Data.
DEVTPW CV DEVTPM CV4 ITYPE2	The Deviation in the True PW, the coefficients for the True PW, and the ratio of the True PM and the card type from the Substrata Statistical Data.
LAT LONG ISW ITRIND ITLIST ITYPE	The latitude longitude, spring/winter wheat indicator, etc., and the card type from segment ID data
ISW1 IYR IMO IDA ISW2 IYR1 IMO1 IDA1	Origins of three arrays dimensioned 8 in which the crop windows are stored by year, month, and day. For spring and winter, from substrata crop window data
DELTWE IPLNTE ISEGSD DELWE1 IPLNT1 ISGSD1	Data from crop calendar error data for spring and winter.

Output: IPRNT - Card image print flag set = 1 if not already 1 to indicate that the pair of card images were printed out.

GOOD1 - Flag set = 1 to indicate that the Substrata Historical Data or the sample segment ID data is no good.

GOOD2 - Flag set = 1 to indicate that the Substrata Statistical Data or the Sample Segment Crop Window Data is no good.

IERR - The error count is advanced by 1 each time ERRMES is called.

FATAL- Set = 1 if there are inconsistencies between the Substrata Historical Data and the Substrata Statistical Data or between the Sample Segment ID Data and the Sample Segment Crop Window Data.

Subroutines Called: None

Error Messages: The type of card containing the error is printed out followed by the card data. T' n one of the following error messages is printed.

1. INCONSISTENT SUBSTRATA HISTORICAL AND STATISTICAL DATA - The country, region, zone, strata, and/or substrata fields do not agree on the indicated cards. Possibly, cards are out of order or missing on one of the input files.

2. INCONSISTENT CROP WINDOW AND CROP WINDOW ERROR DATA - The segment, country, region, zone, strata, and/or substrata fields do not agree on the indicated cards. Possibly, cards are out of order or missing on one of the input files.

3. SUBSTRATA AREA MUST BE POSITIVE - The substrata area is negative or zero or unspecified. It must be > 0.

4. HISTORICAL PROPORTION OF WHEAT MUST BE NON-NEGATIVE - HISTPW is negative. It must be ≥ 0.

5. DEVTPW MUST BE LESS THAN 9.999 IN MAGNITUDE - DEVTPW is > 9.999.

6. COEFF. OF VARIATIONS MUST BE BETWEEN 0.0 AND 9.999 - One of the quantities CV_i is < 0 or > 9.999.

7. DEVTPM MUST BE BETWEEN 0.0 AND 9.999 - DEVTPM < 0 or > 9.999.

8. LATITUDE MUST BE LESS THAN OR EQUAL TO 65.0 DEG - The magnitude of the segment latitude is greater than SLATMX, the maximum allowable latitude.

9. LONGITUDE MUST BE BETWEEN -180 AND +180 DEG. - The magnitude of the segment longitude is greater than 180°.

10. SPRING/WINTER WHEAT INDICATOR MUST BE W OR S - The quantity ISW is unspecified or is a character other than W or S.

11. CV4 MUST BE BETWEEN 0.0 AND 9.999.

12. YEAR MUST BE GREATER THAN 64 - One of the crop window years is < 64.

13. MONTH MUST BE BETWEEN 1 AND 12 - One of the crop window months is ≤ 0 or > 12.

14. DAY MUST BE BETWEEN 1 AND 31. - One of the crop window days is < 1 or > 31.

15. OVERLAPPING CROP WINDOWS - The start date for one of the four crop windows is > the stop date or the stop date of one crop window is ≥ the start date of the next window.

16. GROUP NUMBER MUST BE EQUAL TO 1, 2 OR 3. IGRP must have one of the three values specified above.

17. DELTA ERRORS IN BIO WINDOW PREDICTION DATES ARE NOT BETWEEN -1 AND 1. DELTWE and DELWE1 must be between -1 and +1.

18. SEGMENT SPECIFIED AS ORDINARY MUST HAVE A TRAINING SEGMENT PRIORITY LIST - Each ordinary segment must have at least 1 priority training segment specified.

19. NEITHER NAGR OR NAL CAN BE ZERO.

Definitions of Local Variables: I - Index in a DO loop.

Subroutine CALGEN

Purpose:

To read, check and merge the crop calendar and crop error data card files into one intermediate file for further processing. On Option A listing of the input card files is given.

Input:

ISCW, ICWE, ICALIN, ILIST, IOUT, ENDATA, ICTRY, IREG, IZONE, ISTRA, ISUB, IYR, IMO, IDAY, ITYPE, ICTRY2, IZONE 2, ISTRA2, ISUB2, DELTWE, IPLNTE, ISEGSD, ITYPE2, FATAL

Output:

NCRCD, NCRECD, NCROP, IERR, IREGL, IZONEL, ISTRL, IPRNT, GOOD1, GOOD2

Linkage:

Call CALGEN

Subroutines Used:

None

Local Variable Description

ICALST	=	0 - If for each crop calendar data card, a crop error card is to be read
	=	1 - If current crop error card is to be used for all substrata in a strata
	=	2 - All strata within a zone
	=	3 - All zone within a region
	=	4 - All region within a country
ISKSSC	-	Flag controlling reading of the crop calendar error data
	=	1 - Skip reading the crop calendar error data until the crop calendar data catches up (i.e., until the region, zone, strata, and substrata agree unless the region, zone, strata or substrata from the crop error are 0
ISCC	-	Flag to denote whether crop calendar error file was read after crop calendar data $\neq 0$ - yes, $=0$ - no

ICTEM

IRTEM

IZTEM

ISTEM

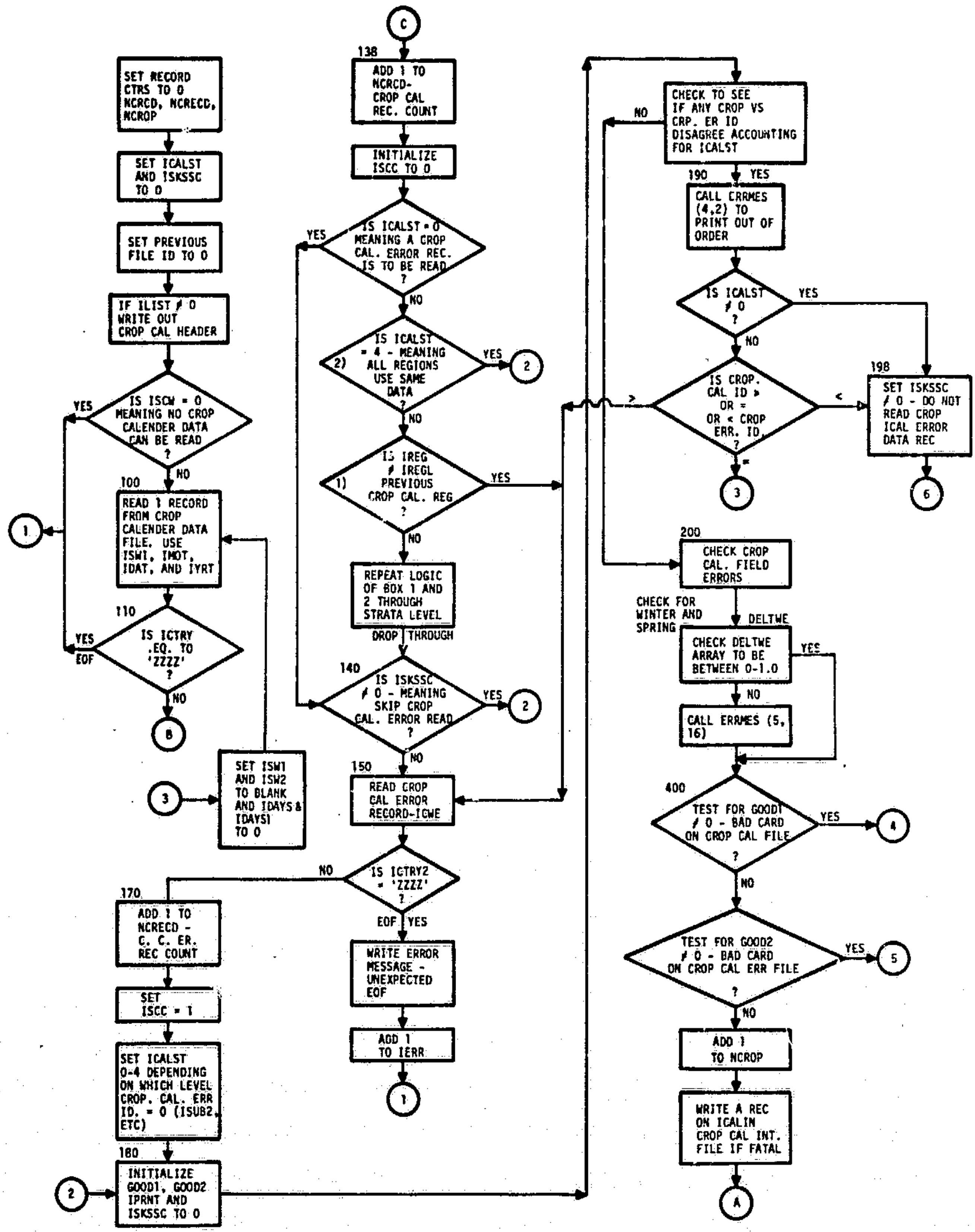
ISSTEM

IMOT(8), IDAT(8), IYRT(8)

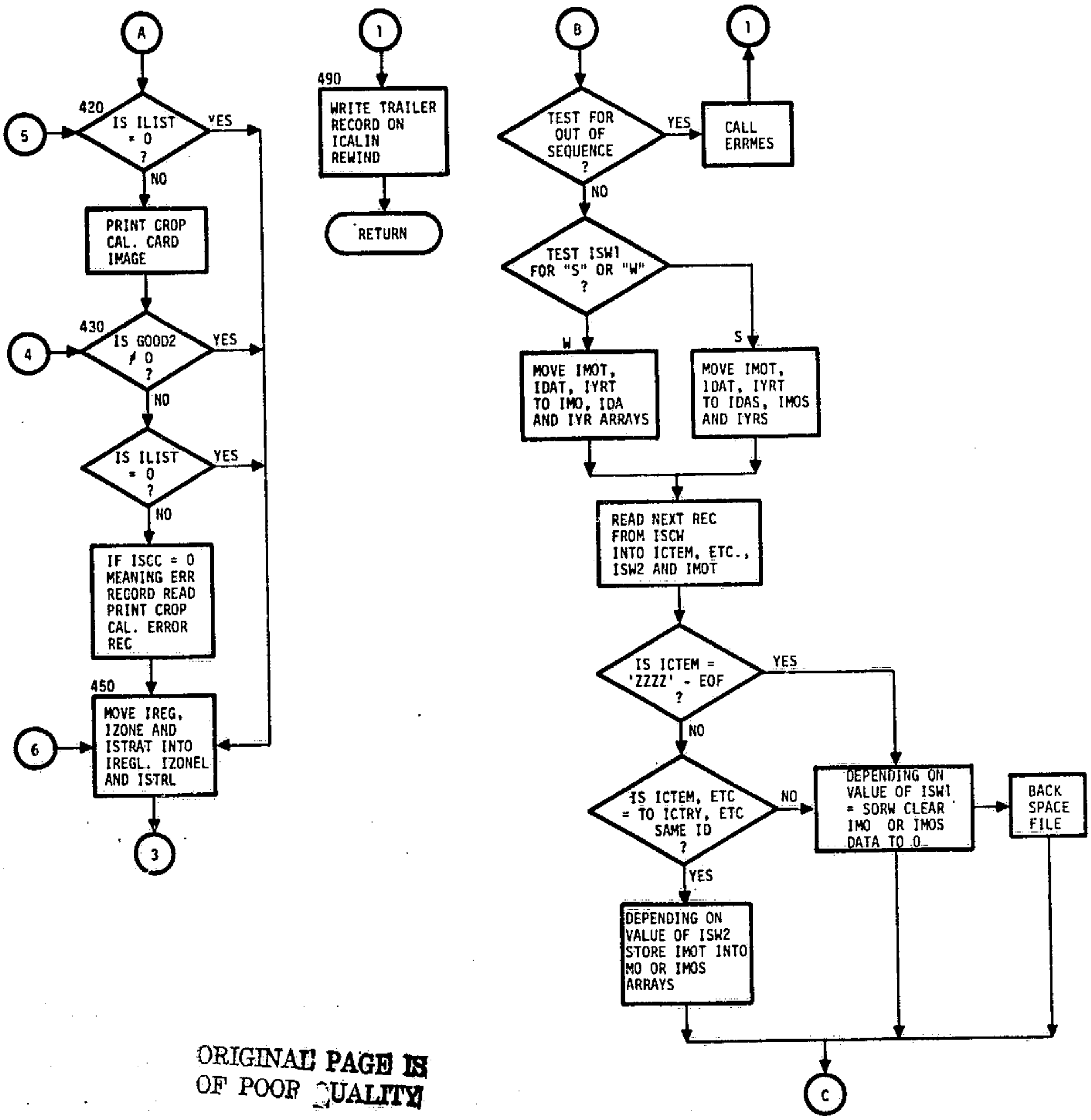
Processing:

See flowchart.

CALGEN FLOWCHART



CALGEN FLOWCHART (Continued)



ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE CALERR

Purpose: Given the reference crop calendar, error terms and standard deviation of calendar error, this routine computes the predicted bio-window start and stop dates and their associated errors.

Input: None through COMMON
INPUT COMMON
IDAYS, IDAYS1, DELTWE, DELWE1, IPLNTE, IPLNT1,
IDAYS - $(t_o, t_f)_i$ pairs $1 \le i \le 4$

Output: CALEND COMMON
IPREDW, IPREDS, ICRPEW, ICRPES

Linkage: Call CALERR (IDAYS, DELTWE, IPLNTE, IPREDW, ICRPEW)

Subroutines Used: None

Local Variables:

$$\text{DELIN1} = \delta_1 (t_{f_1} - t_{o_1}) + .5$$

$$\text{DELIN2} = \delta_2 (t_{f_2} - t_{o_2}) + .5$$

$$\text{DELIN2} = \delta_3 (t_{f_3} - t_{o_3}) + .5$$

Processing:

1. Process data

   a. If IDAYS(1) is 0, then there is no crop calendar data. Skip steps b-d.

   b. Compute DELIN1, DELIN2, DELIN3

   c. Compute predicted dates $(t_o, t_s)_i$ $1 \le i \le 4$

```
IPREDW(1) = IDAYS(1) + IPLNTE
IPREDW(2) = IDAYS(2) + DELIN1 + IPLNTE
IPREDW(3) = IDAYS(3) + DELIN1 + IPLNTE
IPREDW(4) = IDAYS(4) + DELIN1 + DELIN2 + IPLNTE
IPREDW(5) = IDAYS(5) + DELIN1 + DELIN2 + IPLNTE
IPREDW(6) = IDAYS(6) + DELIN1 + DELIN2 + DELIN3 + IPLNTE
IPREDW(7) = IDAYS(7) + DELIN1 + DELIN2 + DELIN3 + IPLNTE
IPREDW(8) = IDAYS(8) + DELIN1 + DELIN2 + DELIN3
            + DELTWE(4) * (IDAYS(8) - IDAYS(7)) + IPLNTE + .5
```

d. **Compute error terms**

ICRPEW(1) = IPLNTE
ICRPEW(2) = IPREDW(2) - IDAYS(2)
ICRPEW(3) = IPREDW(4) - IDAYS(4)
ICRPEW(4) = IPREDW(6) - IDAYS(6)
ICRPEW(5) = IPREDW(8) - IDAYS(8)

SUBROUTINE BETAD

Purpose: To compute a random number based on the Beta distribution or normal distribution, given a random number seed.

Input: No input from COMMON or files.

Output: No output to COMMON or files.

Linkage: CALL BETAD (SEED, XBAR, SIGMA, XI, IOPT, IER)

Input:	SEED	A double precision random number seed used to get a uniform random number P, $0 < P < 1$
	XBAR	Mean value $\overline{X}$, $0 \le \overline{X} \le 1$
	SIGMA	Standard deviation σ, $0 \le \sigma$
	IOPT	= 0 use Beta distribution ≠ 0 use normal distribution
Output:	XI	Random number based on Beta or normal distribution X_i
	IER	Error flag = 0 no errors = 1 XBAR not in range, $0 \le \overline{X} \le 1$ so was reset within subroutine = 2 SIGMA not in range, $0 \le \sigma \le \overline{X} \sqrt{\frac{1 - \overline{X}}{\overline{X} + \epsilon}}$ so was reset within subroutine, $\epsilon = 10^{-4}$ = 3 Fatal error, XI could not be found within constraints of subroutine; e.g., within 35 iterations via the inverse incomplete beta function method
	SEED	To be used for next call to BETAD (a double precision number)

Subroutines used:

CALL RDM1A (SEED, P) to get uniform random number P
SEED = double precision

CALL IBETAI (X, A, B, P, IER) to get incomplete beta function

Note: IBETAI is algorithm AS 63 Appl. Statist. (1973), Vol. 22, No. 3

SQRT (X)	squareroot
ALOG (X)	exponential
EXP (X)	natural logarithm
ALOG(X)	natural logarithm

Local variables:

A		First Beta parameter
B		Second Beta parameter
BP		Recalculated second Beta parameter
CHK		Normal distribution parameter
DIFF		Accuracy check
DIFF1		Check if XI close to 0
EP		10^{-4}, accuracy of answer
FLAG	INTEGER	Flag to signal XBAR > .5
H		Beta approx. parameter
I		Loop counter
K	REAL	2., method threshold constant
P		Output from RDM1A, f(X) for Beta function
PHI		Limit for iteration of P
PLO		Limit for iteration of P
PO		Output from IBETAI
R		88., Gamma constraint
RN		Normal distribution parameter
SIG		Stores SIGMA, or SIGMAL, for use in routine
SG		10^{-10} check on successive answers in loop
SIGMAL		Upper limit on SIGMA
SIGSQ		SIG * SIG, intermediate calculation
SIGT		Method threshold sigma.
T		Normal distribution parameter
W		Beta approx. parameter
XAVG		XBAR, or 1 - XBAR if XBAR > .5
XHI		Limit for iteration of X
XLO		Limit for iteration of X
XSQ		XBAR * XBAR, intermediate calculation
Y		Beta approx. parameter
YP		Beta approx. parameter

BETAD Subroutine Equations

Equation set 1 - normal distribution parameters:

$$T = \begin{cases} \sqrt{\ln \frac{1}{P^2}} & 0 < P \leq .5 \\ \sqrt{\ln \frac{1}{(1-P)^2}} & .5 < P < 1 \end{cases}$$

$$CHK = T - \frac{2.30753 + .27061T}{1. + .99229T + .04481T^2}$$

$$RN = \begin{cases} - CHK & 0 < P \leq .5 \\ + CHK & .5 < P < 1 \end{cases}$$

Equation set 2 - SIGMA upper limit:

$$SIGMAL = XBAR \sqrt{\frac{1 - XBAR}{XBAR + EP}} \qquad EP = 10^{-4}$$

Equation set 3 - Beta function parameters:

$$A = \frac{XBAR^2 - XBAR\,(XBAR^2 + SIGMA^2)}{SIGMA^2}$$

$$B = \left(\frac{1 - XBAR}{XBAR}\right) A$$

Equation set 4 - method threshold:

$$SIGT = XBAR \sqrt{\frac{1 - XBAR}{XBAR + K}} \qquad K = 2$$

Equation set 5 - Beta approximation parameters:

$$YP = -RN$$

$$H = 2 \left(\frac{1}{2A - 1} + \frac{1}{2B - 1} \right)^{-1}$$

$$Y = \frac{YP^2 - 3}{6}$$

$$W = \frac{YP\,(H + Y)^{1/2}}{H} - \left(\frac{1}{2B - 1} - \frac{1}{2A - 1}\right) \left(Y + \frac{5}{6} - \frac{2}{3H}\right)$$

Equation set 6 - XI for Beta approximation:

$$XI = \frac{A}{A + B \cdot e^{2W}} \quad ABS\,(ALOG\,(B) + 2 * W) \leq 87$$

Equation set 7 - recompute A and B, Beta parameters:

$$B^1 = \frac{B}{A + B} \cdot (R - 1)$$

$$A^1 = \frac{BP}{B} \cdot A$$

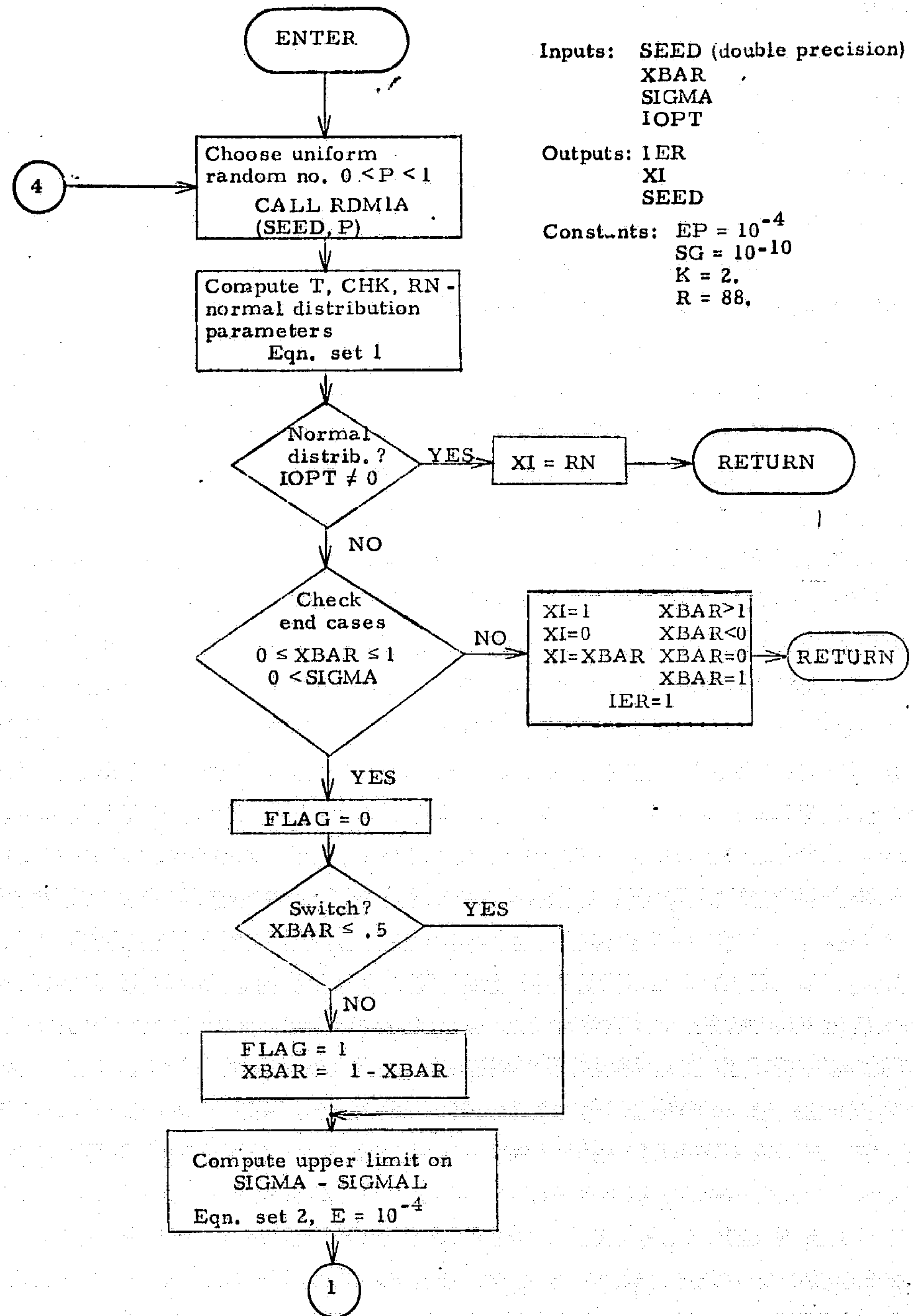
Equation set 8 -

$$XI = \frac{(XH1 + XL0)}{2.}$$

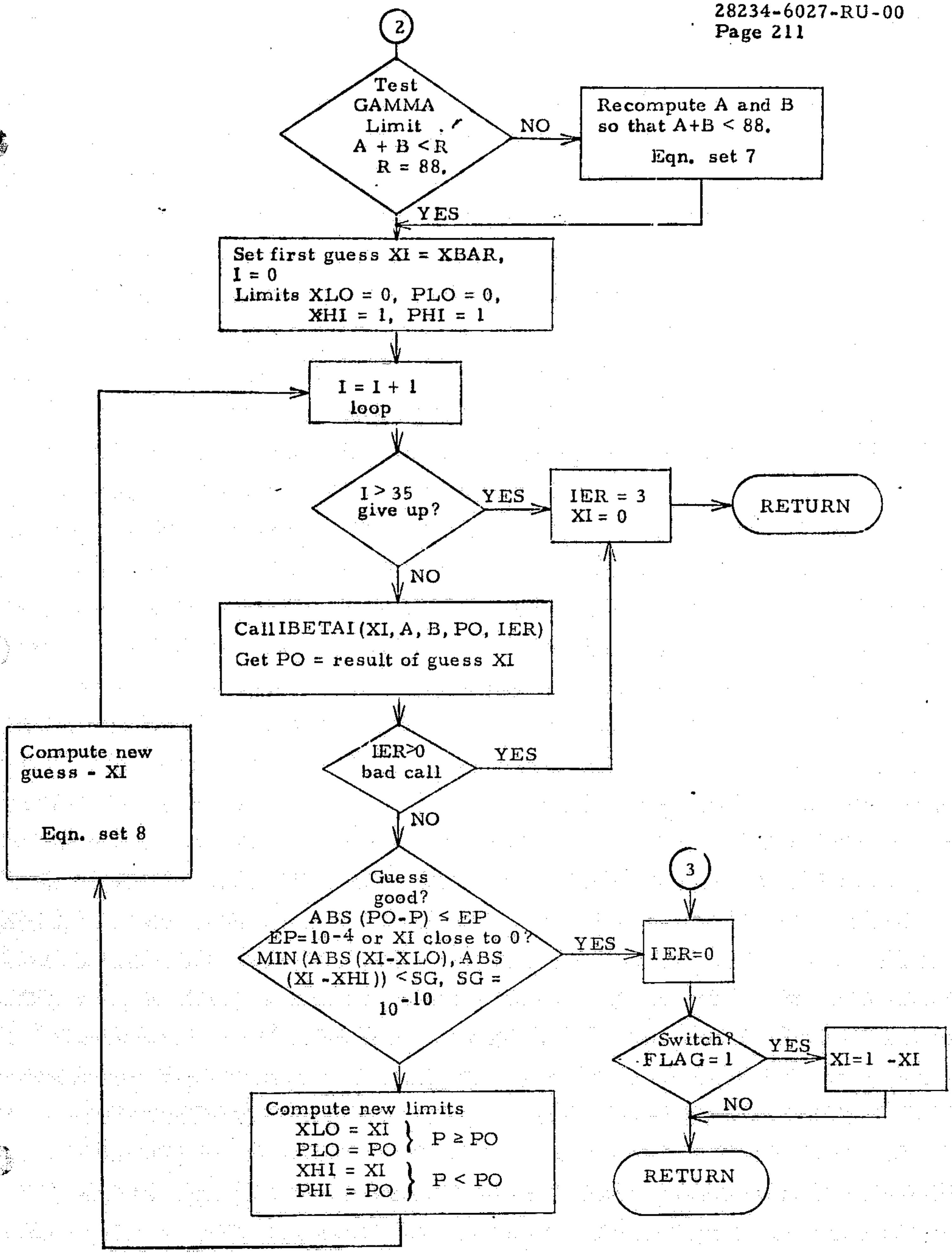
Inputs: SEED (double precision)
XBAR
SIGMA
IOPT

Outputs: IER
XI
SEED

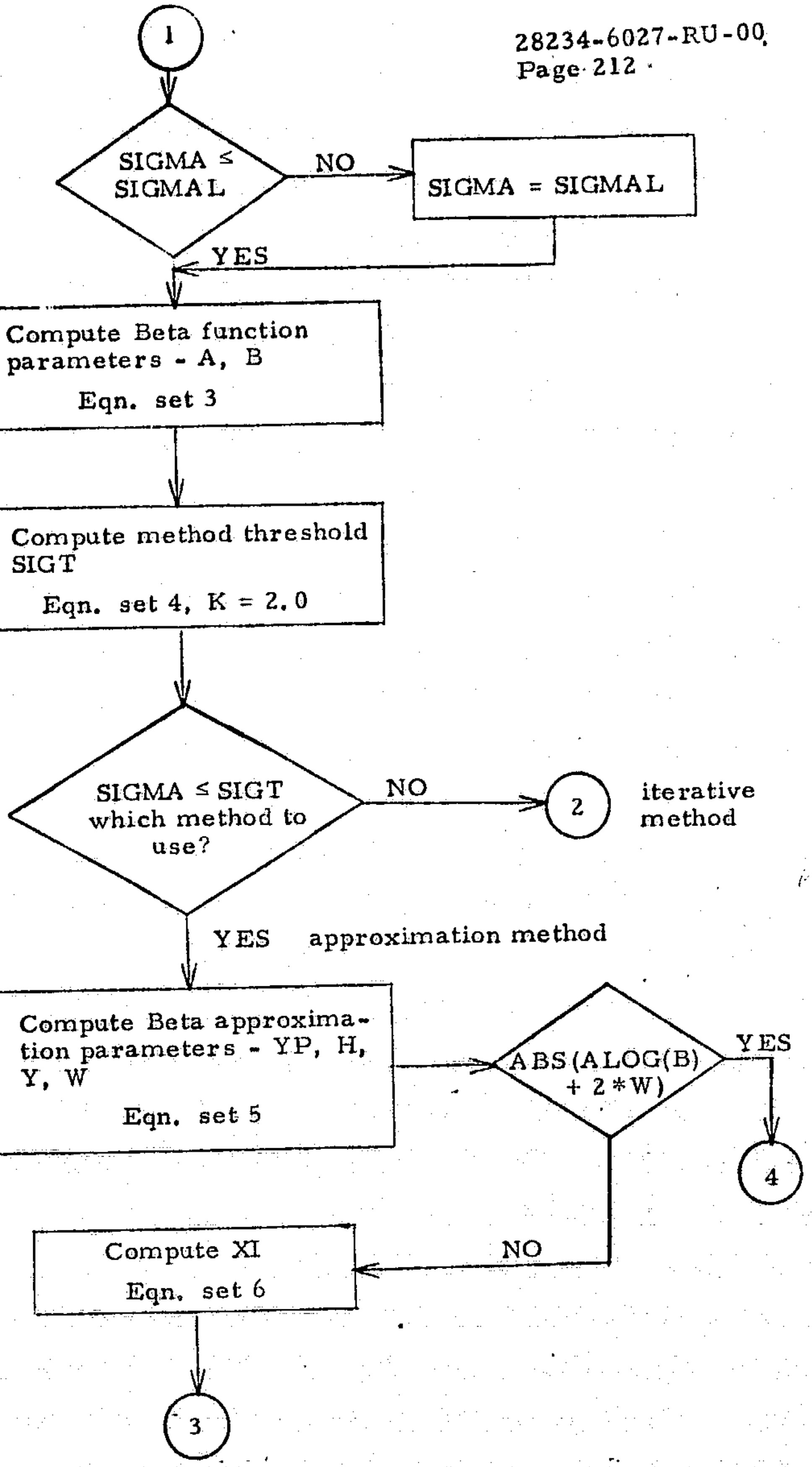
Constants: $EP = 10^{-4}$
$SG = 10^{-10}$
K = 2.
R = 88.



BETAD Subroutine Flowchart (Sheet 1 of 3)



BETAD Subroutine Flowchart (Sheet 3 of 3)



BETAD Subroutine Flowchart (Sheet 2 of 3)

SUBROUTINE GRID

Purpose: To control the determination of the index number for a given segment latitude and longitude on the global index matrix.

Input: QLAT and OLONG in INPUT COMMON
IREG, IZONE, ISTRAT, ISUB and ISEG
IERT in calling sequence

Output: Index in input COMMON

Linkage: Call GRID (IERT)

IERT - Output error flag
=0 - o.k.
≠0 - no index could be found

Subroutines Used:

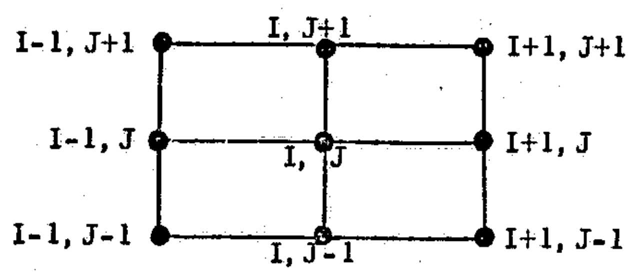
Call GRDCAL (IGRID, JGRID, INDEX, FI, FJ, IERT)
IGRID
JGRID Global grid coordinates
INDEX - Index no.
FI, FJ - Flt. pt. version of IGRID, JGRID
IERT - =0 - o.k., ≠0 - no index found

Processing:

1. Set IERT = 0.
2. Compute IGRID, JGRID, FI and FJ from Equation set 1-6 in Problem Description.
3. Call GRDCAL to get index.
4. If IERT = 0, return.
   If IERT ≠ 0, print error message as defined in Problem Description and return.

SUBROUTINE GRDCAL

Purpose: To determine the most appropriate index value associated with the global grid coordinates IGRID and JGRID. The IGRID and JGRID may not have an index value assigned to it on the index matrix file. If this is the case a boundary grid is drawn about the IGRID and JGRID position and each of the eight grid points is tested in a priority order for an index value. If no index can be found, the subroutine returns control. See Gene Rasmussen's write-up "Weather Index Generation" for details of theory and method. The boundary grid looks like:



where I ≡ IGRID
J ≡ JGRID

Input:
Index from file IGRDTP via COMMON DIRAC
IGRDTP in FLAG COMMON
IGRID, JGRID in CALL SEQ.

Output: Index in CALL SEQ.

Linkage:
CALL GRDCAL (IGRID, JGRID, INDEX, FI, FD, IER)
IGRID, JGRID - input global grid coordinates
INDEX - Index no. (output)
IER - =0 - index found
≠0 - no index found
FI, FJ - floating point equivalents of IGRID, JGRID

Subroutines Used:

CALL RDGRID (JCOOR) - JCOOR is column coordinate ≡ rec. no.

Local Variables:

ISTATE(8, 8)	State matrix; given the state no. (the column no.), each row gives the sequence to try each coordinate of the boundary grid about IGRID, JGRID. This is a constant matrix containing boundary matrix positions.
IBOUND(2, 8)	Boundary coordinate list. Each column of the matrix represents a grid position on the boundary grid matrix. Each row represents the X, Y coordinate and each entry is the amount to subtract or add to IGRID, JGRID to get current grid position. Used in conjunction with ISTATE.
ISTAT	State no., SLOPE - SLOPE
IPT(8)	Current states grid position sequence list

Processing:

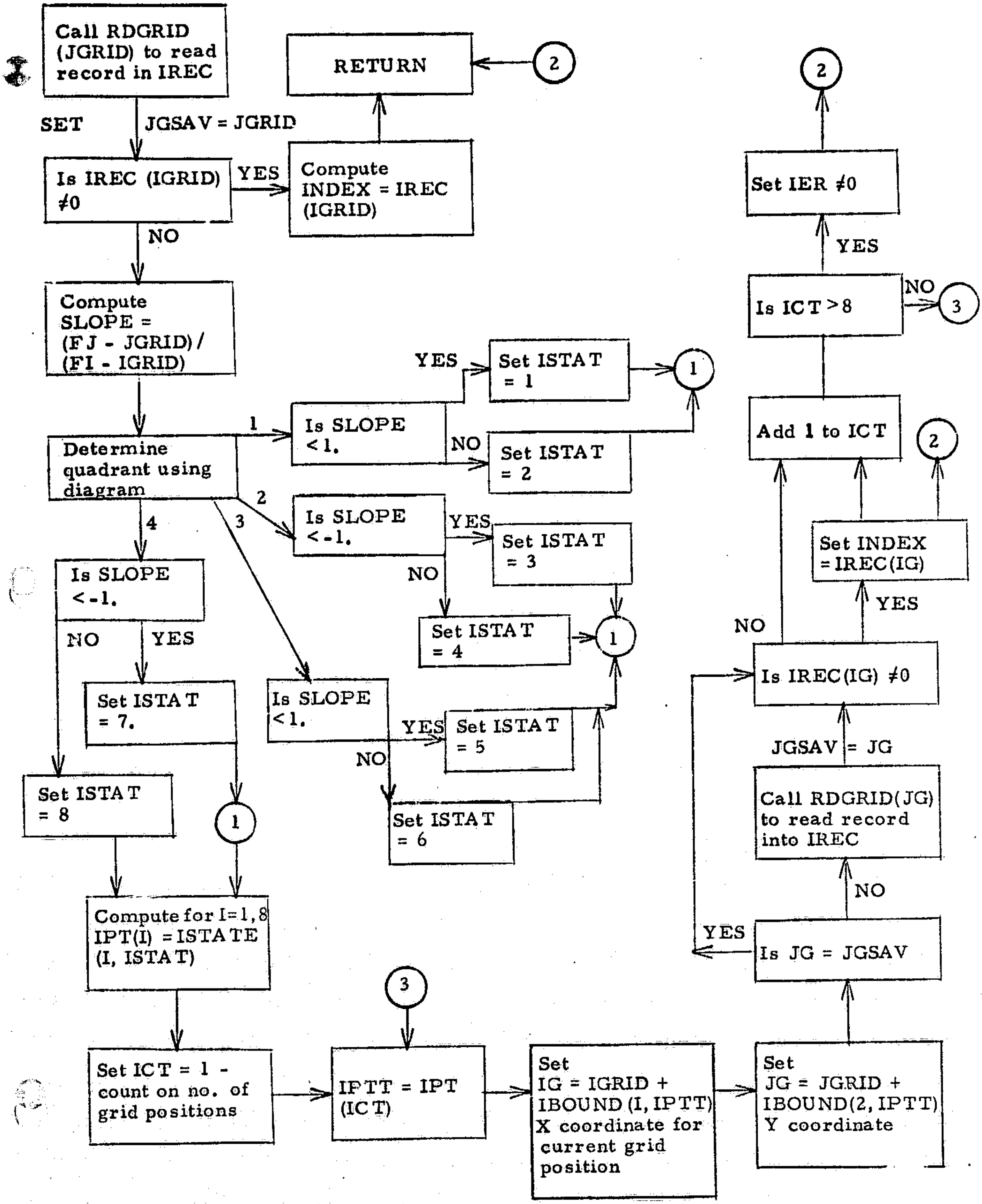
Rules for quadrant determination in boundary matrix.

(FI - IGRID)	+	-	-	+
(FJ - JGRID)	+	+	-	-
Quadrant	1	2	3	4

Rules for determining state given quadrant and SLOPE = $\frac{FJ - JGRID}{FI - IGRID}$

Quadrant	1	1	2	2	3	3	4	4
SLOPE	<1	≥1	<-1	≥-1	<1	≥1	<-1	≥ -1
State	1	2	3	4	5	6	7	8

See flowchart.


Call RDGRID (JGRID) to read record in IREC
SET JGSAV = JGRID
RETURN
2
Is IREC (IGRID) ≠0
YES
Compute INDEX = IREC (IGRID)
NO
Compute SLOPE = (FJ - JGRID) / (FI - IGRID)
Determine quadrant using diagram
1
2
3
4
Is SLOPE < 1.
YES
Set ISTAT = 1
NO
Set ISTAT = 2
Is SLOPE < -1.
YES
Set ISTAT = 3
NO
Set ISTAT = 4
Is SLOPE < -1.
NO
YES
Set ISTAT = 7.
Set ISTAT = 8
Is SLOPE < 1.
YES
Set ISTAT = 5
NO
Set ISTAT = 6
Compute for I=1,8 IPT(I) = ISTATE (I, ISTAT)
Set ICT = 1 - count on no. of grid positions
3
IPTT = IPT (ICT)
Set IG = IGRID + IBOUND (I, IPTT) X coordinate for current grid position
Set JG = JGRID + IBOUND(2, IPTT) Y coordinate
Is JG = JGSAV
YES
NO
Call RDGRID(JG) to read record into IREC
JGSAV = JG
Is IREC(IG) ≠0
YES
NO
Set INDEX = IREC(IG)
Add 1 to ICT
Is ICT > 8
NO
YES
Set IER ≠0

SUBROUTINE RDGRID

Purpose: To initialize, read a record or close the index matrix file on unit IGRDTP

Input: IGRDTP in FLAG COMMON

JG in calling seq.

ICDIN in DIRAC COMMON

Output: One record in IREC in DIRAC COMMON

Linkage: CALL RDGRID(JG)

JG is record no. or flag

If JG = 0, open the file via OPENMS, call for CDC and define file for UNIVAC

If JG = -n, close the file via CALL CLOSEMS for CDC and close for UNIVAC

Processing:

For JG = +n

Read n^{th} record from IGRDTP via CALL READMS for CDC or read for UNIVAC into IREC. 500 word record.

PART V

SUBROUTINE LISTINGS

FLT ALGAMA,1,760430, 54500 , 1

```
      SUBROUTINE ALGAMA(X,GAMMA,IER)                                    ALGAMA
C       CALCULATES THE GAMMA FUNCTION                                   ALGAMA
C                                                                       ALGAMA
C   X       INPUT                                                       ALGAMA
C   GAMMA   OUTPUT                                                      ALGAMA
C   IER     ERROR FLAG = 60 IF X NOT IN RANGE 0. TO 88.                 ALGAMA
C                                                                       ALGAMA
      DIMENSION B(14),C(6)                                              ALGAMA
      DOUBLE PRECISION  B,C,F,G,HLNPI,Y,Z
      DATA B/-3.0301908102800-6,2.7983288993830-5,-1.21417348706320-4,  ALGAMA  *NEW
     *3.7556605226307D-4,-8.3756468513517D-4,2.0610918502554D-3,        ALGAMA  *NEW
     *-2.8360625203728D-3,1.1149714335778D-2,-2.6618659495306D-4,       ALGAMA  *NEW
     *7.4248915419444D-2,8.1576926124155D-2,4.1184033016678D-1,         ALGAMA  *NEW
     *4.2278433510233D-1,9.9999999999999D-1/                                    *NEW
      DATA C/-1.9175269175269D-3,8.4175084175084D-4,-5.9523809523809D   ALGAMA  *NEW
     *-4,7.9365079365079D-4,-2.7777777777778D-3,8.3333333333333D-2/     ALGAMA  *NEW
      DATA HLNPI/.91893853320467/                                       ALGAMA  **-7
      GAMMA=0.                                                          ALGAMA
      IER=60                                                            ALGAMA
      IF (X .LE. 0.0 .OR. X .GT. 88.0) RETURN                           ALGAMA
      IER=0                                                             ALGAMA
      Y=X                                                               ALGAMA
      IF(Y.GE.10.) GO TO 60                                             ALGAMA
      Z=DMOD(Y,1.0)
      IF(Z.NE.0.0) GO TO 20                                             ALGAMA
      Z=1.0                                                             ALGAMA
      G=2.0                                                             ALGAMA
      GO TO 27                                                          ALGAMA
   20 G=B(1)                                                            ALGAMA
      DO 26 J=2,14                                                      ALGAMA
   26 G=G*Z+B(J)                                                        ALGAMA
   27 IF(Y-3.0) 28,45,38                                                ALGAMA
   28 IF(Y.GT.2.0) GO TO 45                                             ALGAMA
      G=G/Y                                                             ALGAMA
      IF(Y.LE.1.0) G=G/(Y+1.0)                                          ALGAMA
      GO TO 45                                                          ALGAMA
   38 F=1.0                                                             ALGAMA
      K = DINT(Y-7+.5)-2                                                        *NEW
      DO 40 J=1,K                                                       ALGAMA  **-1
      Y=Y-1.0                                                           ALGAMA
   40 F=F*Y                                                             ALGAMA
      G=G*F                                                             ALGAMA
   45 GAMMA=DLOG(G)
      RETURN                                                            ALGAMA
   60 Z=Y*Y                                                             ALGAMA
      G=C(1)                                                            ALGAMA
      DO 65 J=2,6                                                       ALGAMA
   65 G=G/Z+C(J)                                                        ALGAMA
      G=((G/Y+HLNPI)+((Y-.5)*DLOG(Y)-Y))
      GAMMA=G                                                           ALGAMA
      RETURN                                                            ALGAMA
      END                                                               ALGAMA
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
 * ELT BETAD,1,760401, 67415     , 1

      SUBROUTINE BETAD(SEED,XBAR,SIGMA,XI,IOPT,IER)                    BETAD
      DOUBLE PRECISION SEED                                            BETAD
      INTEGER FLAG                                                     BETAD
      REAL K                                                           BETAD
      DATA R/88./                                                      BETAD
      DATA K/2./                                                       BETAD
      DATA EP/.00005/                                                  BETAD
      DATA SG /1.E-8 /                                                 BETAD
      DATA FLAG/0/                                                     BETAD
      DATA I/0/                                                        BETAD
   50 CONTINUE                                                         MODS
      FLAG =0                                                          MODR
      I = 0                                                            MODR
      XAVG=XBAR                                                        BETAD
      XI=0.0                                                           BETAD
      IER=0                                                            BETAD
C                                                                      BETAD
C  CHOOSE UNIFORM RANDOM NUMBER                                        BETAD
      CALL RDM1A(SEED,P)                                               BETAD
C                                                                      BETAD
C COMPUTE T,CHK,RN, - NORMAL DISTRIB. PARAMETERS                       BETAD
      T=SQRT(ALOG(1.0/(P*P)))                                          BETAD
      IF(P.GT.0.5)T=SQRT(ALOG(1.0/((1.0-P)*(1.0-P))))                  BETAD
      CHK=T-(2.30753+.27061*T)/(1.0+0.99229*T+.04481*T*T)              BETAD
      RN=CHK                                                           BETAD
      IF(P.LE.0.5) RN=-RN                                              BETAD
C                                                                      BETAD
C  IF NORMAL DISTRIB. OPTION, JUMP OUT                                 BETAD
      IF(IOPT.EQ.0) GO TO 70                                           BETAD
      XI=RN                                                            BETAD
      RETURN                                                           BETAD
   70 CONTINUE                                                         BETAD
C                                                                      BETAD
C  CHECK FOR END CASES                                                 BETAD
      IF(XBAR.LT.0. .OR.XBAR.GT.1.) IER=1                              BETAD
      IF(SIGMA.LT.0.0) IER=2                                           BETAD
      IF (XBAR.GT.0.0.AND.XBAR.LT.1.0.AND.SIGMA.GT.0.0) GO TO 10       BETAD
      XI=XBAR                                                          BETAD
      IF(XBAR .LT. 0.)XI = 0.0                                         MODS
      IF(XBAR .GT. 1.)XI = 1.0                                         MODS
      RETURN                                                           BETAD
C                                                                      BETAD
C SWITCH IF AVERAGE ABOVE HALF                                         BETAD
   10 CONTINUE                                                         BETAD
      IF (XBAR.LE.0.5) GO TO 20                                        BETAD
      FLAG=1                                                           BETAD
      XAVG=1.0-XBAR                                                    BETAD
C                                                                      BETAD
C  COMPUTE UPPER LIMIT ON SIGMA                                        BETAD
   20 CONTINUE                                                         BETAD
      SIGMAL=XAVG*SQRT((1.00-XAVG)/(XAVG+EP))                          BETAD
      SIG=SIGMA                                                        BETAD
      IF(SIGMA.LE.SIGMAL) GO TO 30                                     BETAD
      SIG=SIGMAL                                                       BETAD
      IER=2                                                            BETAD
C                                                                      BETAD
C  COMPUTE THE METHOD THRESHHOLD                                       BETAD
   30 CONTINUE                                                         BETAD
```

```
      SIGT=XAVG*SQRT((1.0-XAVG)/(XAVG*K))                                BETAD
C                                                                        BETAD
C   COMPUTE BETA PARAMETERS A AND B                                      BETAD
      XSQ=XAVG*XAVG                                                      BETAD
      SIGSQ=SIG*SIG                                                      BETAD
      A=(XSQ-XAVG*(XSQ+SIGSQ))/SIGSQ                                     BETAD
      B=((1.0-XAVG)/XAVG)*A                                              BETAD
      IF(SIG.GT.SIGT) GO TO 40                                           BETAD
C                                                                        BETAD
C  APPROXIMATION METHOD                                                  BETAD
C                                                                        BETAD
C                                                                        BETAD
C  COMPUTE BETA APPROXIMATION PARAMETERS                                 BETAD
      YP=-RN                                                             BETAD
      H=2.0/ (1.0/(2.0*A- 1.0) + 1.0/(2.0*B - 1.0))                      BETAD
      Y=(YP*YP-3.0)/6.0                                                  BETAD
      W=(YP*SQRT(H+Y))/                                                  BETAD
     1  H - (1.0/(2.0*B-1.0)-1.0/(2.0*A-1.0))*(Y+5.0/6.0-2.0/(3.0*H))    BETAD
C                                                                        BETAD
C  COMPUTE XI                                                            BETAD
      IF(ABS(ALOG(B) + 2.*W) .GT. 87.0)GO TO 50                          MOD5
      XI=A/(A+B*EXP(2.0*W))                                              BETAD
      GO TO 60                                                           BETAD
C                                                                        BETAD
C  ITERATIVE METHOD                                                      BETAD
C                                                                        BETAD
C RECOMPUTE A AND B IF OVER LIMIT                                        BETAD
   40 CONTINUE                                                           BETAD
      CHK=A+B                                                            BETAD
      IF(CHK.LT.R) GO TO 80                                              BETAD
      BP=(B/CHK)*(R-1.)                                                  BETAD
      A=(BP/B)*A                                                         BETAD
      B=BP                                                               BETAD
   80 CONTINUE                                                           BETAD
      I=0                                                                BETAD
      PHI=1.                                                             BETAD
      PLO=0.                                                             BETAD
      XHI=1.                                                             BETAD
      XLO=0.                                                             BETAD
      XI=XAVG                                                            BETAD
  120 I=I+1                                                              BETAD
      IF(I.GT.35) GO TO 100                                              BETAD
      CALL IBETAI(XI,A,B,PO,IER)                                         BETAD
      IF(IER.GT.0) GO TO 100                                             BETAD
      DIFF=ABS(PO-P)                                                     BETAD
      DIFF1=AMIN1(ABS(XI-XLO),ABS(XI-XHI))                               BETAD
      IF(DIFF.LE.EP.OR.DIFF1.LT.SG ) GO TO 60                            BETAD
C                                                                        BETAD
C  CHECK IF DONE, WITHIN TOLERANCE                                       BETAD
      IF(PO.LT.P) GO TO 105                                              BETAD
      XHI=XI                                                             BETAD
      PHI=PO                                                             BETAD
      GO TO 106                                                          BETAD
  105 CONTINUE                                                           BETAD
      XLO=XI                                                             BETAD
      PLO=PO                                                             BETAD
  106 CONTINUE                                                           BETAD
      XI = (XHI + XLO)/2.                                                MOD9
      GO TO 120                                                          BETAD
C ERROR RETURN                                                           BETAD
```

```
100 CONTINUE                                                    BFTAD
    IER=5                                                       BFTAD
    XI=0.0                                                      BFTAD
    RETURN                                                      BFTAD
 60 CONTINUE                                                    BFTAD
    IF(FLAG.EQ.1) XI=1.0 - XI                                   BFTAD
    RETURN                                                      BFTAD
    END                                                         BFTAD
```

```
@ELT CALERR,1,760401, 67397    , 1
      SUBROUTINE CALERR(IDAYS,DELTWE,IPLNTE,IPREDW,ICRPEW)          CALERR
C     COMPUTES PREDICTED BIO WINDOW START AND STOP DATES WITH THEIR  CALERR
C     ASSOCIATED ERRORS.                                            CALERR
      DIMENSION IDAYS(8),DELTWE(4),IPREDW(8),ICRPEW(5)              CALERR
      IF(IDAYS(1) .EQ. 0)GO TO 50                                   CALERR
      DELIN1 = DELTWE(1)*(IDAYS(2) - IDAYS(1)) + .5                 CALERR
      DELIN2 = DELTWE(2)*(IDAYS(4) - IDAYS(3)) + .5                 CALERR
      DELIN3 = DELTWE(3)*(IDAYS(6) - IDAYS(5)) + .5                 CALERR
      IPREDW(1) = IDAYS(1) + IPLNTE                                 CALERR
      IPREDW(2) = IDAYS(2) + DELIN1 + IPLNTE                        CALERR
      IPREDW(3) = IDAYS(3) + DELIN1 + IPLNTE                        CALERR
      IPREDW(4) = IDAYS(4) + DELIN1 + DELIN2 + IPLNTE               CALERR
      IPREDW(5) = IDAYS(5) + DELIN1 + DELIN2 + IPLNTE               CALERR
      IPREDW(6) = IDAYS(6) + DELIN1 + DELIN2 + DELIN3 + IPLNTE      CALERR
      IPREDW(7) = IDAYS(7) + DELIN1 + DELIN2 + DELIN3 + IPLNTE      CALERR
      IPREDW(8) = IDAYS(8) + DELIN1 + DELIN2 + DELIN3 + DELTWE(4)   CALERR
     1*(IDAYS(8) - IDAYS(7)) + .5 + IPLNTE                          MOD3
      ICRPEW(1) = IPLNTE                                            CALERR
      ICRPEW(2) = IPREDW(2) - IDAYS(2)                              CALERR
      ICRPEW(3) = IPREDW(4) - IDAYS(4)                              CALERR
      ICRPEW(4) = IPREDW(6) - IDAYS(6)                              CALERR
      ICRPEW(5) = IPREDW(8) - IDAYS(8)                              CALERR
   50 RETURN                                                        CALERR
      END                                                           CALERR
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
@ ELT CALGEN,1,760401, 67395   , 1
      SUBROUTINE CALGEN                                                 CALGEN
C     THIS ROUTINE READS,CHECKS AND MERGES THE CROP CALENDER AND CROP   CALGEN
C     CALENDER AND CROP ERROR DATA CARD FILES INTO AN INTERMEDIATE FILE.CALGEN
      COMMON /CONST/                                                    CONST
     1PT,DEGTOR,      SLATMX,ENDATA,ZERO,INP,IOUT,INISUB,INTSEG,ISIDF,  CONST
     2ICHF,ISHF,LRSIDF,LRCHF,LRSHF,IWIN,MXSEG,ICALIN,ICHL,AREACF        CONST
     3,ISW1,ISW2,IGRDTP                                                 MOD8
      INTEGER ZERO                                                      CONST
      COMMON /FLAGS/                                                    FLAGS
     1IERR,IPRNT,GOOD1,NSHC,NSSC,NSFGC,NSUBR,NSFGR,NRSIDF,NRCHF,NRSHF,  FLAGS
     2ISSC,ISTAT,IREGL,IZONEL,ISTRL,ISUBL,ISEGL,NSEG,IMXSEG,NTLIST,     FLAGS
     3READSF,GOOD2,FATAL,IDAYS(8),IDSEG(150)                            FLAGS
     1,IMOT(8),IDAT(8),IYRT(8),IYR1(8),IMO1(8),IDA1(8)                  MOD2
      INTEGER GOOD1,GOOD2,READSF,FATAL                                  FLAGS
      COMMON /INPUT/                                                    INPUT
     1RSLEN,SUBAR,HISTPR,DEVTPR,CV(3),DEVTPR,OLAT,OLONG,SLAT(3),SLONG(3)INPUT
     2,NAMSUB(2),ICTRY,ICTRY2,LEASE,ILIST,ITSFG,ISHD,ISSD,IS10,ISCR,    INPUT
     3ISEG,IREG,IZONE,ISTRAT,ISUB,ISW,ITYPE,ITYPE2,INDEX,ISEG2,IREG2,   INPUT
     4IZONE2,ISTRA2,ISUB2,LAT(3),LONG(3),IYR(8),IMO(8),IDA(8),CV4,IGRP, INPUT
     5DELTHF(4),IPLNIT,ISFGSD,ITRIND,NAGR,NAL,ITLIST(6),NCRED,NCHECD,   INPUT
     6NCROP,IDAYS1(8),DFLKET(4),IPLNT1,ISGSD1,IDAS(8),IMOS(8),IYRS(8)   INPUT
      DOUBLE PRECISION RSFED                                            INPUT
      REAL NAMSUB,ICTRY,ICTRY2                                          INPUT
      DATA IBK/1H /                                                     MOD2
      REAL ICTEM                                                        CALGEN
      NCRED = 0                                                         CALGEN
      NCHECD = 0                                                        CALGEN
      NCROP = 0                                                         CALGEN
      ICALST = 0                                                        CALGEN
      ISKSSF = 0                                                        CALGEN
      IREGL = 0                                                         CALGEN
      IZONEL = 0                                                        CALGEN
      ISUBL = 0                                                         MOD4
      ISTRL = 0                                                         CALGEN
      IF(ILIST .NE. 0)WRITE(IOUT,10)                                    CALGEN
   10 FORMAT(50H1  SUBSTRATA CROP CALENDER AND CALENDER ERROR DATA/)    CALGEN
      IF(ISCR .EQ. 0)GO TO 490                                          CALGEN
  100 ISW1 = IBK                                                        MOD2
      ISW2 = IBK                                                        MOD2
      DO 105 I=1,8                                                      MOD2
      IDAYS(I) = 0                                                      CALGEN
      IDAYS1(I) = 0                                                     CALGEN
  105 CONTINUE                                                          CALGEN
      READ(ISCR,1)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISW1,(IYRT(I),IMOT(I),   CALGEN
     1IDAT(I),I=1,8),ITYPE                                              CALGEN
    1 FORMAT(5X,A4,2I4,2I5,A1,3(6I2,1X),6I2,I1)                         CALGEN
  110 IF(ICTRY .EQ. ENDATA)GO TO 490                                    CALGEN
      IF(IREG    -  IREGL)480,111,114                                   CALGEN
  111 IF(IZONE   -  IZONEL)480,112,114                                  MOD4
  112 IF(ISTRAT  -  ISTRL)480,113,114                                   MOD4
  113 IF(ISUB    -  ISUBL)480,480,114                                   MOD4
  114 CONTINUE                                                          MOD4
      IF(ISW1 .EQ. 1HS)GO TO 120                                        MOD4
      DO 115 I=1,8                                                      CALGEN
      IDA(I) = IDAT(I)                                                  CALGEN
      IMO(I) = IMOT(I)                                                  CALGEN
      IYR(I) = IYRT(I)                                                  CALGEN
  115 CONTINUE                                                          CALGEN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                                                                      CALGEN
      GO TO 125                                                       CALGEN
  120 DO 122 I=1,8                                                    CALGEN
      IDAS(I) = IDAT(I)                                               CALGEN
      IMOS(I) = IMOT(I)                                               CALGEN
      IYRS(I) = IYRT(I)                                               CALGEN
  122 CONTINUE                                                        CALGEN
  125 READ(ISCR,1)ICTEM,IRTEM,IZTEM,ISTEM,ISSTEM,ISW2,(IYR1(I),       MOD2
     1IMO1(I),IDA1(I),I=1,8),ITYPE1                                   MOD2
      IF(ICTEM .EQ. ENDATA)GO TO 130                                  CALGEN
      IF(ICTEM .NE. ICTRY)GO TO 130                                   CALGEN
      IF(IRTEM .NE. IREG) GO TO 130                                   CALGEN
      IF(IZTEM .NE. IZONE)GO TO 130                                   MOD5
      IF(ISTEM .NE. ISTRAT)GO TO 130                                  CALGEN
      IF(ISSTEM .NE. ISUB)GO TO 130                                   CALGEN
      IF(ISW2 .EQ. IHS)GO TO 128                                      CALGEN
      DO 126 I=1,8                                                    CALGEN
      IDA (I) = IDA1(I)                                               MOD2
      IMO(I) = IMO1(I)                                                MOD2
      IYR(I) = IYR1(I)                                                MOD2
  126 CONTINUE                                                        CALGEN
      GO TO 138                                                       CALGEN
  128 DO 129 I=1,8                                                    CALGEN
      IDAS(I) = IDA1(I)                                               MOD2
      IMOS(I) = IMO1(I)                                               MOD2
      IYRS(I) = IYR1(I)                                               MOD2
  129 CONTINUE                                                        CALGEN
      GO TO 138                                                       CALGEN
  130 ISW2 = IBK                                                      CALGEN
      DO 131 I=1,8                                                    MOD3
      IDA1(I) = 0                                                     MOD3
      IMO1(I) = 0                                                     MOD3
      IYR1(I) = 0                                                     MOD3
  131 CONTINUE                                                        MOD3
      ITYPE1 = 0                                                      MOD3
      IF(ISW1 .EQ. IHW)GO TO 133                                      MOD3
      DO 132 I=1,8                                                    CALGEN
      IDA(I) = 0                                                      CALGEN
      IMO(I) = 0                                                      CALGEN
      IYR(I) = 0                                                      CALGEN
  132 CONTINUE                                                        CALGEN
      GO TO 135                                                       CALGEN
  133 DO 134 I=1,8                                                    CALGEN
      IDAS(I) = 0                                                     CALGEN
      IMOS(I) = 0                                                     CALGEN
      IYRS(I) = 0                                                     CALGEN
  134 CONTINUE                                                        CALGEN
  135 BACKSPACE ISCR                                                  CALGEN
  138 NCRCD = ICRCD + 1                                               CALGEN
      ISCC = 0                                                        CALGEN
      IF(ICALST .EQ. 0)GO TO 140                                      CALGEN
      IF(ICALST .EQ. 4)GO TO 180                                      CALGEN
      IF(IREG .NE. IREGL)GO TO 150                                    CALGEN
      IF(ICALST .EQ. 3)GO TO 180                                      CALGEN
      IF(IZONE .NE. IZONEL)GO TO 150                                  CALGEN
      IF(ICALST .EQ. 2)GO TO 180                                      CALGEN
      IF(ISTRAT .EQ. ISTRL)GO TO 180                                  CALGEN
  140 IF(ISRSSL .NE. 0)GO TO 180                                      CALGEN
  150 READ(IEPL,2)ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,(DELTHE(I),I=1,4), CALGEN
     1IPLHIF,ISGSD,(PELHEI(I),I=1,4),IPLNEI,ISGSD1,ITYPE2              CALGEN
    2 FORMAT(5X,A4,2I4,2I5,2(4F5.3,I3,I2),2X,I1)                      MOD5
```

```
      IF(ICTRY2 .NE. ENDATA)GO TO 170                                   CALGEN
  165 WRITE(IOUT,21)                                                    CALGEN
   21 FORMAT(//44H UNEXPECTED EOF ON CROP CALENDER ERROR FILE.)         MOD3
      IERR = IERR + 1                                                   CALGEN
      GO TO 490                                                         CALGEN
  170 NCRECD = NCRECD + 1                                               CALGEN
      ISEG = 1                                                          CALGEN
      ICALST = 0                                                        CALGEN
      IF(ISUB2 .EQ. 0) ICALST = 1                                       CALGEN
      IF(ISTRA2 .EQ. 0)ICALST = 2                                       CALGEN
      IF(IZONE2 .EQ. 0)ICALST = 3                                       CALGEN
      IF(IREG2  .EQ. 0)ICALST = 4                                       CALGEN
  180 GOOD1 = 0                                                         CALGEN
      GOOD2 = 0                                                         CALGEN
      IPRNT = 0                                                         CALGEN
      ISKSSG = 0                                                        CALGEN
      IF(ICTRY .NE. ICTRY2)GO TO 190                                    CALGEN
      IF(ICALST .EQ. 4)GO TO 200                                        CALGEN
      IF(IREG .NE. IREG2)GO TO 190                                      CALGEN
      IF(ICALST .EQ. 3)GO TO 200                                        CALGEN
      IF(IZONE .NE. IZONE2)GO TO 190                                    CALGEN
      IF(ICALST .EQ. 2)GO TO 200                                        CALGEN
      IF(ISTRAT .NE. ISTRA2)GO TO 190                                   CALGEN
      IF(ICALST .EQ. 1)GO TO 200                                        CALGEN
      IF(ISUB .EQ. ISUB2)GO TO 200                                      CALGEN
  190 CALL ERRMES(4,2)                                                  CALGEN
      IF(ICALST .EQ. 4)GO TO 198                                        CALGEN
      IF(IREG - IREG2)198,192,150                                       CALGEN
  192 IF(ICALST .EQ. 3)GO TO 198                                        CALGEN
      IF(IZONE - IZONE2)198,194,150                                     CALGEN
  194 IF(ICALST .EQ. 2)GO TO 198                                        CALGEN
      IF(ISTRAT - ISTRA2)198,196,150                                    CALGEN
  196 IF(ISUB - ISUB2)198,100,150                                       CALGEN
  198 ISKSSG = 1                                                        CALGEN
      GO TO 450                                                         CALGEN
  200 IPRNT = 0                                                         CALGEN
  205 IF(IYR(1) .EQ. 0)GO TO 225                                        CALGEN
      DO 250 I=1,8                                                      CALGEN
      IF(IYR(I) .LT. 64)CALL ERRMES(4,12)                               CALGEN
      IF(IMO(I) .LT. 1 .OR. IMO(I) .GT. 12)CALL ERRMES(4,13)            CALGEN
      IF(IDA(I) .LT. 1 .OR. IDA(I) .GT. 31)CALL ERRMES(4,14)            CALGEN
      FEDA = IDA(I)                                                     CALGEN
      CALL LFTACF(FDA,IMO(I),IYR(I),DAYS)                               CALGEN
      IDAYS(I) = DAYS                                                   CALGEN
      IF(I  .EQ. 1)GO TO 250                                            CALGEN
      IF(MOD(I,2) .EQ. 1)GO TO 240                                      CALGEN
      IF(IDAYS(I-1) .GT. IDAYS(I))CALL ERRMES(4,15)                     CALGEN
      GO TO 250                                                         CALGEN
  240 IF(IDAYS(I-1) .GE. IDAYS(I))CALL ERRMES(4,15)                     CALGEN
  250 CONTINUE                                                          CALGEN
  225 IF(IYRS(1) .EQ. 0)GO TO 255                                       CALGEN
      DO 235 I=1,8                                                      CALGEN
      IF(IYRS(I).LT. 64)CALL ERRMES(4,12)                               CALGEN
      IF(IMOS(I) .LT. 1 .OR. IMOS(I) .GT. 12)CALL ERRMES(4,13)          CALGEN
      IF(IDAS(I) .LT. 1 .OR. IDAS(I) .GT. 31)CALL ERRMES(4,14)          CALGEN
      FLDA1 = IDAS(I)                                                   CALGEN
      CALL LFPACF(FLDA1,IMOS(I),IYRS(I),DAYS1)                          CALGEN
      IDAYS1(I) = DAYS1                                                 CALGEN
      IF(I  .EQ. 1)GO TO 235                                            CALGEN
      IF(MOD(I,2) .EQ. 1)GO TO 230                                      CALGEN
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      IF(IDAYS1(I-1) .GT. IDAYS1(I))CALL ERRMES(4,15)                   CALGEN
      GO TO 235                                                         CALGEN
  230 IF(IDAYS1(I-1) .GE. IDAYS1(I))CALL ERRMES(4,15)                   CALGEN
  235 CONTINUE                                                          CALGEN
  255 IF(ISEC .EQ. 0)GO TO 400                                          CALGEN
      IPRNT = 0                                                         CALGEN
      DO 260 I=1,4                                                      CALGEN
      IF(DELTWE(I) .LT. -1.0 .OR. DELTWE(I) .GT. 1.0)CALL ERRMES(5,17)  MODS
      IF(DELWEI(I) .LT. -1.0 .OR. DELWEI(I) .GT. 1.0)                   MODS
     1  CALL ERRMES(5,17)                                               CALGEN
  260 CONTINUE                                                          CALGEN
  400 IF(GOOD1 .NE. 0)GO TO 430                                         CALGEN
      IF(GOOD2 .NE. 0)GO TO 420                                         CALGEN
      IF(FATAL .NE. 0)GO TO 420                                         MOD2
      NCROP = NCROP + 1                                                 CALGEN
      IF(FATAL .EQ. 0)WRITE(ICALIN)ICTRY,IREG,IZONE,ISTRAT,ISUB,(IDAYS(ICALGEN
     1),I=1,8),(DELTWE(I),I=1,4),IPLNTE,ISEGSD,(IDAYS1(I),I=1,8),       CALGEN
     2(DELWEI(I),I=1,4),IPLNTI,ISGSD1                                   CALGEN
  420 IF(ILIST .EQ. 0)GO TO 450                                         CALGEN
      WRITE(IOUT,11)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISW1,(IYPT(I),IMOT(I), MOD2
     1IDAT(I),I=1,8),ITYPE,ISW2,(IYR1(I),IMO1(I),IDA1(I),I=1,8),ITYPE1  MOD2
   11 FORMAT(6X,A4,2I4,2I5,A1,3(6I2,1X),6I2,I1/26X,A1,3(6I2,1X),6I2,I1) CALGEN
  430 IF(GOOD2 .NE. 0 .OR. ILIST .EQ. 0)GO TO 450                       CALGEN
      IF(ISEC .NE. 0)WRITE(IOUT,12)ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,    CALGEN
     1(DELTWE(I),I=1,4),IPLNTE,ISEGSD,(DELWEI(I),I=1,4),IPLNTI,ISGSD1,  CALGEN
     2ITYPE2                                                            CALGEN
   12 FORMAT(6X,A4,2I4,2I5,2(4F5.3,I3,I2),2X,I1)                        MODS
  450 IREGL = IREG                                                      CALGEN
      IZONEL = IZONE                                                    CALGEN
      ISTRL = ISTRAT                                                    CALGEN
      ISUBL = ISUB                                                      MOD4
      GO TO 100                                                         CALGEN
  480 CALL ERRMES(4,2)                                                  MOD4
  490 WRITE(ICALIN) ENDATA,(ZERO,I=1,32)                                CALGEN
      REWIND ICALIN                                                     CALGEN
      RETURN                                                            CALGEN
      END                                                               CALGEN
```

```
@ ELT ERRMES,1,760002, 49900   , 1

      SUBROUTINE ERRMES (ICARD,ICODE)                                   ERRMES
C        ERROR MESSAGE SUBROUTINE                                       ERRMES
C                                                                       ERRMES
      COMMON /CONST/                                                    CONST
     1PI,DEGTOR,      SLATMX,ENDATA,ZERO,INP,IOUT,INTSUR,INTSEG,ISIDF,  CONST
     2ICHF,ISHF,LRSIDF,LRCHF,LRSHF,IWIN,MXSEG,ICALIN,ICHE,AREACF        CONST
     3,ISW1,ISW2,IGRDIP                                                 MOD8
      INTEGER ZERO                                                      CONST
      COMMON /FLAGS/                                                    FLAGS
     1IERR,IPRNT,GOOD1,NSHC,NSSC,NSEGC,NSUBR,NSEGR,NRSIDF,NRCHF,NRSHF,  FLAGS
     2ISSC,ISTAT,IREGL,IZONEL,ISTRL,ISUBL,ISEGL,NSEG,IMXSEG,NTLIST,     FLAGS
     3READSF,GOOD2,FATAL,IDAYS(8),IDSEG(150)                            FLAGS
     1,IMOI(8),IDAT(8),IYRT(8),IYR1(8),IMO1(8),IDA1(8)                  MOD2
      INTEGER GOOD1,GOOD2,READSF,FATAL                                  FLAGS
      COMMON /INPUT/                                                    INPUT
     1RSEED,SUBAR,HISTPW,DEVTPW,CV(3),DEVTPP,QLAT,QLONG,SLAT(3),SLONG(3)INPUT
     2,NAMSUB(2),ICTRY,ICTRY2,ICASE,ILIST,ITSEG,ISHD,ISSD,ISID,ISCH,    INPUT
     3ISEG,IREG,IZONE,ISTRAT,ISUB,ISW,ITYPE,ITYPE2,INDEX,ISEG2,IREG2,   INPUT
     4IZONE2,ISTRA2,ISUB2,LAT(3),LONG(3),IYR(8),IMO(8),IDA(8),CV4,IGRP, INPUT
     5DELTWF(4),IPLNIF,ISLCSD,ITRIND,NAGR,NAL,ITLIST(6),NCRCD,NCRFCD,   INPUT
     6NCROP,IDAYST(8),DELREI(4),IPLNTI,ISGSPI,IDAS(8),IMOS(8),IYRS(8)   INPUT
      DOUBLE PRECISION RSEED                                            INPUT
      REAL NAMSUB,ICTRY,ICTRY2                                          INPUT
C                                                                       ERRMES
C                                                                       ERRMES
C                                                                       ERRMES
      IF ( IPRNT .NE. 0 )  GO TO 100                                    ERRMES
      WRITE (IOUT,800)                                                  ERRMES
  800 FORMAT (//24H ERROR IN THE FOLLOWING )                            ERRMES
      GO TO (10,20,30,40,50), ICARD                                     ERRMES
   10 WRITE (IOUT,801)  ICTRY,IREG,IZONE,ISTRAT,ISUB,NAMSUB,SUBAR,HISTPWERRMES
     1,NAGR,NAL,IGRP,ITYPE                                                       *NEW
      GOOD1= 1                                                          ERRMES   **-1
  801 FORMAT (31H SUBSTRATA HISTORICAL DATA CARD /                      ERRMES
     16X,A4,2I4,2I5,,2A4,F10.0,F7.3,2I4,I1,18X,I1/)                     MOD5
      IF ( ICODE .NE. 1 )  GO TO 90                                     ERRMES
      FATAL= 1                                                          ERRMES
   20 WRITE (IOUT,802) ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,DEVTPW,CV,DEVTPPERRMES
     1,CV4,ITYPE2                                                       MOD5
  802 FORMAT (32H SUBSTRATA STATISTICAL DATA CARD /                     ERRMES
     16X,A4,2I4,2I5,F6.3,5F5.3,21X,I1/)                                 MOD5
      GOOD2= 1                                                          ERRMES
      GO TO 90                                                          ERRMES
   30 WRITE (IOUT,803)  ISEG,ICTRY,IREG,IZONE,ISTRAT,ISUB,LAT,LONG      ERRMES
     1  ,ISW,ITRIND,(ITLIST(I),I=1,6),ITYPE                             ERRMES
  803 FORMAT (28H SAMPLE SEGMENT ID DATA CARD /1X,I4                    ERRMES
     1,1X,A4,2I4,2I5,1X,A1,I2,1X,I2,1X,A1,I3,1X,I2,1X,A1,1X,I1,,1X,6I4  ERRMES
     1,8X,I1/)                                                          ERRMES
      GOOD1= 1                                                          ERRMES
      GO TO 90                                                          MOD2
   40 FATAL = 1                                                         MOD2
      WRITE(IOUT,804)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISW1                  MOD2
     1,(IYRT(I),IMOT(I),IDAT(I),I=1,8)                                  MOD2
      WRITE(IOUT,804)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISW2,                 MOD2
     1(IYR1(I),IMO1(I),IDA1(I),I=1,8)                                   MOD2
  804 FORMAT (31H 1 OR 2 CROP WINDOW DATA CARDS /                       ERRMES
     1  1X      ,A4,2I4,2I5,A1,6I2,3(1X,6I2),I1/)                       ERRMES
      GOOD1= 1                                                          ERRMES
```

ORIGINAL PAGE IS OF POOR QUALITY

28234-6027-RU-00
Page 228

```
      IF(ICODE .NE. 2)GO TO 90                                          MOD2
   50 WRITE(IOUT,805)       ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,            ERRMES
     1(DELTWE(I),I=1,4),IPLNTF,ISEGSD,(DELWF1(I),I=1,4),IPLNT1,ISGSD1   ERRMES
      GOOD2=1                                                           ERRMES
  805 FORMAT(29H CROP WINDOW ERROR DATA CARDS /                         ERRMES
     11X,A4,1X,I3,I4,2I5,4F5.3,I3,I2,4F5.3,I3,I2)                       MOD5
   90 IPRNT= 1                                                          ERRMES
  100 IERR= IERR + 1                                                    ERRMES
      GO TO (110,120,130,140,150,160,170,180,190,200,210,220,230,240    ERRMES
     1  ,250,260,270,280,290), ICODE                                    ERRMES
  110 WRITE (IOUT,901)                                                  ERRMES
  901 FORMAT(55H INCONSISTENT SUBSTRATA HISTORICAL AND STATISTICAL DATA ERRMES
     1  /)                                                              ERRMES
      GO TO 999                                                         ERRMES
  120 WRITE (IOUT,902)                                                  ERRMES
  902 FORMAT(52H INCONSISTENT CROP WINDOW AND CROP WINDOW ERROR DATA/)  ERRMES
      GO TO 999                                                         ERRMES
  130 WRITE ( IOUT,903)                                                 ERRMES
  903 FORMAT(33H SUBSTRATA AREA MUST BE POSITIVE./)                     ERRMES
      GO TO 999                                                         ERRMES
  140 WRITE (IOUT,904)                                                  ERRMES
  904 FORMAT(51H HISTORICAL PROPORTION OF WHEAT MUST BE NONNEGATIVE/)   ERRMES
      GO TO 999                                                         ERRMES
  150 WRITE (IOUT,905)                                                  ERRMES
  905 FORMAT(45H DEVIPH MUST BE LESS THAN 9.999 IN MAGNITUDE./)         ERRMES
      GO TO 999                                                         ERRMES
  160 WRITE (IOUT,906)                                                  ERRMES
  906 FORMAT(51H COEFF. OF VARIATIONS MUST BE BETWEEN 0.0 AND 9.999/)   ERRMES
      GO TO 999                                                         ERRMES
  170 WRITE (IOUT,907)                                                  ERRMES
  907 FORMAT(37H DEVIPM MUST BE BETWEEN 0.0 AND 9.999/)                 ERRMES
      GO TO 999                                                         ERRMES
  180 WRITE (IOUT,908)                                                  ERRMES
  908 FORMAT(49H LATITUDE MUST BE LESS THAN OR EQUAL TO 65.0 DEG./)     ERRMES
      GO TO 999                                                         ERRMES
  190 WRITE (IOUT,909)                                                  ERRMES
  909 FORMAT(45H LONGITUDE MUST BE BETWEEN -180 AND +180 DEG./)         ERRMES
      GO TO 999                                                         ERRMES
  200 WRITE (IOUT,910)                                                  ERRMES
  910 FORMAT(46H SPRING/WINTER WHEAT INDICATOR MUST BE  W OR S/)        ERRMES
      GO TO 999                                                         ERRMES
  210 WRITE (IOUT,911)                                                  ERRMES
  911 FORMAT(34H CV4 MUST BE BETWEEN 0.0 AND 9.999/)                    MOD3
      GO TO 999                                                         ERRMES
  220 WRITE (IOUT,912)                                                  ERRMES
  912 FORMAT(29H YEAR MUST BE GREATER THAN 64/)                         ERRMES
      GO TO 999                                                         ERRMES
  230 WRITE (IOUT,913)                                                  ERRMES
  913 FORMAT(31H MONTH MUST BE BETWEEN 1 AND 12/)                       ERRMES
      GO TO 999                                                         ERRMES
  240 WRITE (IOUT,914)                                                  ERRMES
  914 FORMAT(29H DAY MUST BE BETWEEN 1 AND 31/)                         ERRMES
      GO TO 999                                                         ERRMES
  250 WRITE (IOUT,915)                                                  ERRMES
      GO TO 999                                                         ERRMES
  260 WRITE(IOUT,916)                                                   ERRMES
  916 FORMAT(35H GROUP NO MUST BE EQUAL TO 1,2 OR 3/)                   ERRMES
      GO TO 999                                                         ERRMES
  270 WRITE(IOUT,917)                                                   ERRMES
  917 FORMAT(67H DELTA ERRORS IN BIO WINDOW PREDICTION DATES ARE NOT BETERRMES
```

```
      1WELN-1 OR 1/)                                                 MODS
       GO TO 999                                                     ERRMES
  280 WRITE(IOUT,918)                                                ERRMES
  918 FORMAT(73H SEGMENT SPECIFIED AS ORDINARY MUST HAVE A TRAINING SEGMERRMES
     1ENT PRIORITY LIST/)                                            ERRMES
       GO TO 999                                                     ERRMES
  290 WRITE(IOUT,919)                                                ERRMES
  919 FORMAT(32H NEITHER NAGR OR NAL CAN BE ZERO/)                   ERRMES
  915  FORMAT(25H OVERLAPPING CROP WINDOWS/)                         ERRMES
  999  RETURN                                                        ERRMES
       END                                                           ERRMES
```

ORIGINAL PAGE IS
OF POOR QUALITY

* ELT FILGEN,1,760402, 49895 , 1

```
      SUBROUTINE FILGEN                                                  FILGEN
C        READS DATA FROM 4 INPUT FILES AND GENERATES 2 INTERMEDIATE      FILGEN
C        DISK FILES.                                                     FILGEN
C        THE INPUT FILES ARE ...                                         FILGEN
C           1. SUBSTRATA HISTORICAL DATA                                 FILGEN
C           2. SUBSTRATA STATISTICAL DATA                                FILGEN
C           3. SAMPLE SEGMENT ID DATA                                    FILGEN
C           4. SAMPLE SEGMENT CROP WINDOW DATA                           FILGEN
C                                                                        FILGEN
C        THE OUTPUT FILES ARE ...                                        FILGEN
C           1. SUBSTRATA DATA                                            FILGEN
C           2. SAMPLE SEGMENT DATA                                       FILGEN
C                                                                        FILGEN
      COMMON /CONST/                                                     CONST
     1PI,DEGTOR,      SLATMX,ENDATA,ZERO,INP,IOUT,INTSUR,INTSEG,ISIDF,   CONST
     2ICWF,ISHF,LRSIDF,LRCWF,LRSHF,IWIN,MXSEG,ICALIN,ICWE,AREACF         CONST
     3,ISW1,ISW2,IGRDTP                                                  MOD4
      INTEGER ZERO                                                       CONST
C                                                                        FILGEN
      COMMON /FLAGS/                                                     FLAGS
     1IERR,IPRNT,GOOD1,NSHC,NSSC,NSEGC,NSUBR,NSEGR,NRSIDF,NRCWF,NRSHF,   FLAGS
     2ISSC,ISTAT,IREGL,IZONEL,ISTRL,ISUBL,ISEGL,NSEG,IMXSEG,NTLIST,      FLAGS
     3READSF,GOOD2,FATAL,IDAYS(8),IDSEG(150)                             FLAGS
     1,IHOT(8),IDAT(8),IYRT(8),IYR1(8),IMO1(8),IDA1(8)                   MOD2
      INTEGER GOOD1,GOOD2,READSF,FATAL                                   FLAGS
C                                                                        FILGEN
      COMMON /INPUT/                                                     INPUT
     1RSEED,SUBWN,HISTPN,DEVTPN,CV(3),DEVTPM,OLAT,OLONG,SLAT(3),SLONG(3) INPUT
     2,NAMSUB(2),ICTRY,ICTRY2,ICASE,ILIST,ITSEG,ISHD,ISSD,ISID,ISCN,     INPUT
     3ISEG,IREG,IZONE,ISTRAT,ISUB,ISW,ITYPE,ITYPE2,INDEX,ISEG2,IREG2,    INPUT
     4IZONE2,ISTRA2,ISUB2,LAT(3),LONG(3),IYR(8),IMO(8),IDA(8),CV4,IGRP,  INPUT
     5DELTWE(8),IPLNTF,ISEGSD,ITRIND,NAGR,NAL,ITLIST(6),NCRCD,NCRECD,    INPUT
     6NCROP,IDAYS1(8),DELWE1(8),IPLNT1,ISGSD1,IDAS(8),IMOS(8),IYRS(8)    INPUT
      DOUBLE PRECISION RSEED                                             INPUT
      REAL NAMSUB,ICTRY,ICTRY2                                           INPUT
C                                                                        FILGEN
      COMMON /LDATA/                                                     LDATA
     1WDATA(8),SWDATA(8)                                                 LDATA
      INTEGER WDATA,SWDATA                                               LDATA
      COMMON /TRATAB/                                                    TRATAB
     1ITRAIN(2000),ITRPRI(2000)                                          TRATAB
C                                                                        FILGEN
C                                                                        FILGEN
C                                                                        FILGEN
      NSHC= 0                                                            FILGEN
      NSSC= 0                                                            FILGEN
      NSEGL= 0                                                           FILGEN
      NSUBR= 0                                                           FILGEN
      NSEGR= 0                                                           FILGEN
      ISTAT= 0                                                           FILGEN
      IREGL= 0                                                           FILGEN
      IZONEL = 0                                                         FILGEN
      ISTRL= 0                                                           FILGEN
      ISUBL = 0                                                          MOD4
      ISKSSC= 0                                                          FILGEN
      DO 5 I=1,2000                                                      FILGEN
      ITRAIN(I) = 0                                                      FILGEN
      ITRPRI(I) = 0                                                      FILGEN
```

```
    5 CONTINUE                                                        FILGEN
      ILOC = 1                                                        FILGEN
      ILOC1 = 1                                                       FILGEN
      IF ( ILIST .NE. 0 )  WRITE (IOUT,10)                            FILGEN
   10 FORMAT (44H1  SUBSTRATA HISTORICAL AND STATISTICAL DATA/)       FILGEN
C                                                                     FILGEN
      IF ( ISHD .EQ. 0 )  GO TO 490                                   FILGEN
C        READ SUBSTRATA HISTORICAL DATA CARD                          FILGEN
  100 READ (ISHD,1)  ICTRY,IREG,IZONE,ISTRAT,ISUB,NAMSUB,SUBAR,HISTPN FILGEN
     1,NAGR,NAL,IGRP,ITYPE                                            MOD5
    1 FORMAT(5X,A4,2I4,2I5,2A4,F10.0,F7.3,2I4,I1,18X,I1)              MOD5
C                                                                     FILGEN
C        TEST FOR END OF SUBSTRATA HISTORICAL DATA                    FILGEN
C                                                                     FILGEN   **-1
  110 IF ( ICTRY .EQ. ENDATA )  GO TO 490                             FILGEN
      IF(IREG    -   IREGL)480,115,130                                MOD4
  115 IF(IZONE   -  IZONEL)480,120,130                                MOD4
  120 IF(ISTRAT  -   ISTRL )480,125,130                               MOD4
  125 IF(ISUB    -   ISUBL)480,480,130                                MOD4
  130 CONTINUE                                                        MOD4
      IF ( ICTRY .NE. SHUSA .AND. ICTRY .NE. 4HCAND )  ARFACF= 0.01   FILGEN
      NSHC= NSHC + 1                                                  FILGEN
      ISSC= 0                                                         FILGEN
      IF ( ISTAT .EQ. 0 )  GO TO 140                                  FILGEN
C        ISTAT .NE. 0.  LAST SUBSTRATUM STATISTICAL CARD READ HAD ZERO FILGEN
C        FIELD FOR REGION, ZONE, STRATUM, OR SUBSTRATUM               FILGEN
      IF ( ISTAT .EQ. 4 )  GO TO 180                                  FILGEN
      IF ( IREG .NE. IREGL )  GO TO 150                               FILGEN
C        REGION FIELD ON SUBSTRATA HISTORICAL DATA CARD UNCHANGED.    FILGEN
      IF ( ISTAT .EQ. 3 )  GO TO 180                                  FILGEN
      IF ( IZONE .NE. IZONEL )  GO TO 150                             FILGEN
C        ZONE FIELD UNCHANGED                                         FILGEN
      IF ( ISTAT .EQ. 2 )  GO TO 160                                  FILGEN
      IF ( ISTRAT .EQ. ISTRL )  GO TO 180                             FILGEN
C        IF ISKSSC .NE. 0, SKIP READING OF SUBSTRATA STATISTICAL DATA FILGEN
C        CARD.                                                        FILGEN
  140 IF ( ISKSSC .NE. 0 )  GO TO 180                                 FILGEN
C        READ NEXT SUBSTRATA STATISTICAL DATA CARD.                   FILGEN
  150 READ (ISSD,2)  ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,DEVTPH,CV,DEVTPN FILGEN
     1,CV4,ITYPE2                                                     MOD5
    2 FORMAT(5X,A4,2I4,2I5,F6.3,5F5.3,21X,I1)                         MOD5
C                                                                     FILGEN
C        TEST FOR END OF SUBSTRATA STATISTICAL DATA                   FILGEN
  160 CONTINUE                                                        FILGEN   **-1
C                                                                     FILGEN
      IF ( ICTRY2 .NE. ENDATA )  GO TO 170                            FILGEN
  165 WRITE (IOUT,21)                                                 FILGEN
   21 FORMAT(//45H UNEXPECTED EOF ON SUBSTRATA STATISTICAL FILE)      MOD4
      IERR= IERR + 1                                                  FILGEN
      GO TO 490                                                       FILGEN
  170 NSSC= NSSC + 1                                                  FILGEN
      ISSC= 1                                                         FILGEN
C        CHECK SUBSTRATA STATISTICAL DATA CARD FOR ZERO ZONE, STRATUM, FILGEN
C        OR SUBSTRATUM DATA FIELDS AND SET ISTAT FLAG ACCORDINGLY.    FILGEN
      ISTAT= 0                                                        FILGEN
      IF ( ISUB2 .EQ. 0 )  ISTAT= 1                                   FILGEN
      IF ( ISTRA2 .EQ. 0 )  ISTAT= 2                                  FILGEN
      IF ( IZONE2 .EQ. 0 )  ISTAT= 3                                  FILGEN
      IF ( IREG2 .EQ. 0 )  ISTAT= 4                                   FILGEN
C        CHECK FOR CONSISTENCY BETWEEN SUBSTRATA HISTORICAL DATA CARD FILGEN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C         AND SUBSTRATA STATISTICAL DATA CARD.                          FILGEN
 180  GOOD1= 0                                                          FILGEN
      GOOD2= 0                                                          FILGEN
      IPRNT= 0                                                          FILGEN
      ISKSSC= 0                                                         FILGEN
      IF ( ICTRY .NE. ICTRY2 )  GO TO 190                               FILGEN
      IF ( ISTAT .EQ. 4 )  GO TO 200                                    FILGEN
      IF ( IREG .NE. IREG2 )  GO TO 190                                 FILGEN
      IF ( ISTAT .EQ. 3 )  GO TO 200                                    FILGEN
      IF ( IZONE .NE. IZONE2 )  GO TO 190                               FILGEN
      IF ( ISTAT .EQ. 2 )  GO TO 200                                    FILGEN
      IF ( ISTRAT .NE. ISTRA2 )  GO TO 190                              FILGEN
      IF ( ISTAT .EQ. 1 )  GO TO 200                                    FILGEN
      IF ( ISUB .EQ. ISUB2 )  GO TO 200                                 FILGEN
C         SUBSTRATA HISTORICAL DATA AND SUBSTRATA STATISTICAL DATA      FILGEN
C         DO NOT AGREE                                                  FILGEN
 190  CALL ERRMES (1,1)                                                 FILGEN
C         CHECK WHICH FIELD DISAGREED.                                  FILGEN
      IF ( ISTAT .EQ. 4 )  GO TO 198                                    FILGEN
      IF ( IREG - IREG2 )  198,192,150                                  FILGEN
 192  IF ( ISTAT .EQ. 3 )  GO TO 198                                    FILGEN
      IF ( IZONE - IZONE2 )  198,194,150                                FILGEN
 194  IF ( ISTAT .EQ. 2 )  GO TO 198                                    FILGEN
      IF ( ISTRAT - ISTRA2 )  198,196,150                               FILGEN
 196  IF ( ISUB - ISUB2 )  198,100,150                                  FILGEN
C         SET ISKSSC TO READ SUBSTRATA HISTORICAL DATA CARD AND SKIP    FILGEN
C         READING SUBSTRATA STATISTICAL DATA CARD.                      FILGEN
 198  ISKSSC= 1                                                         FILGEN
      GO TO 450                                                         FILGEN
C         CHECK DATA ENTRIES ON SUBSTRATA HISTORICAL DATA CARD.         FILGEN
 200  IF ( SUBAR .GT. 0.0 )  GO TO 210                                  FILGEN
      CALL ERRMES (1,3)                                                 FILGEN
  210 IF(NACR .LE. 0 .AND. NAL .NE. 0)GO TO 220                         MOD2
      CALL ERRMES(1,19)                                                 FILGEN
  220 IF(IGRP .GT. 0 .AND. IGRP .LT. 4)GO TO 222                        MOD5
      CALL ERRMES(2,16)                                                 MOD5
  222 IF(HISTPH .GE. 0.0 .AND. HISTPH .LE. 1.E2)GO TO 300               MOD5
      CALL ERRMES (1,4)                                                 FILGEN
C         IF SUBSTRATA STATISTICAL DATA CARD WAS JUST READ,             FILGEN
C         CHECK DATA ENTRIES.  OTHERWISE SKIP OVER DATA CHECKING        FILGEN
 300  IF ( ISSC .EQ. 0 )  GO TO 400                                     FILGEN
      IPRNT= 0                                                          FILGEN
      IF(ABS(DEVIPH) .LE. 9.999)GO TO 217                               MOD5
      CALL ERRMES (2,5)                                                 FILGEN
  217 IF(CV4 .GE. 0.0 .AND. CV4 .LE. 9.999)GO TO 320                    MOD2
      CALL ERRMES(2,11)                                                 MOD2
 320  DO 330 I=1,3                                                      FILGEN
      IF ( CV(I) .GE. 0.0 .AND. CV(I) .LE. 9.999 )  GO TO 330           FILGEN
      CALL ERRMES (2,6)                                                 FILGEN
 330  CONTINUE                                                          FILGEN
      IF ( DEVIPH .GE. 0.0 .AND. DEVIPH .LE. 9.999 )  GO TO 400         FILGEN
      CALL ERRMES (2,7)                                                 FILGEN
C                                                                       FILGEN
 400  IF ( GOOD1 .NE. 0 )  GO TO 430                                    FILGEN
      IF ( GOOD2 .NE. 0 )  GO TO 420                                    FILGEN
      IF(FATAL .NE. 0)GO TO 420                                         MOD2
C         ALL SUBSTRATA DATA IS VALID.  WRITE ONE RECORD OF SUBSTRATA   FILGEN
C         DATA ON INTERMEDIATE FILE.                                    FILGEN
      NSUBR= NSUBR + 1                                                  FILGEN
C         CONVERT HISTPH TO STRAIGHT PERCENT                            FILGEN
```

```
C         CONVERT UNITS OF AREA                                           FILGEN
      AREA= SUBAR*AREACF                                                  FILGEN
      X = (NAL*10289.712)                                                 MOD4
      RK = AREA/X                                                         MOD4
      Y = (NAGR*RK*10289.712)                                             MOD4
      HSTPW = (HISTPW*AREA)/Y                                             MOD4
      IF ( FATAL .EQ. 0 )                                                 FILGEN
     1  WRITE (INTSUB)  ICTRY,IREG,IZONE,ISTRAT,ISUB,NAMSUB,AREA,HSTPW    FILGEN
     1       ,DEVTPW,CV,DEVTPM,NAGR,NAL,CV4,IGRP                          FILGEN
  420 IF ( ILIST .EQ. 0 )  GO TO 450                                      FILGEN
      WRITE (IOUT,11)  ICTRY,IREG,IZONE,ISTRAT,ISUB,NAMSUB,SUBAR,HISTPW   FILGEN
     1,NAGR,NAL,IGRP,ITYPE                                                MOD5
   11 FORMAT(6X,A4,2I4,2I5,2A4,F10.0,F7.3,2I4,I1,18X,I1)                  MOD5
  430 IF ( GOOD2 .NE. 0 )  GO TO 450                                      FILGEN
      IF ( ILIST .EQ. 0 )  GO TO 450                                      FILGEN
      IF ( ISSG .NE. 0 )  WRITE (IOUT,12)  ICTRY2,IREG2,IZONE2,ISTRA2     FILGEN
     1,ISUB2,DEVTPW,CV,DEVTPM,CV4,ITYPE2                                  MOD5
   12 FORMAT(6X,A4,2I4,2I5,F6.3,5F5.3,21X,I1)                             MOD5
  450 IREGL= IREG                                                         FILGEN
      IZONEL= IZONE                                                       FILGEN
      ISTRL= ISTRAT                                                       FILGEN
      ISUBL = ISUB                                                        MOD4-
      GO TO 100                                                           FILGEN
C                                                                         FILGEN
  480 CALL ERRMES(1,1)                                                    MOD4
C         WRITE TRAILER RECORD ON INTERMEDIATE SUBSTRATA FILE             FILGEN
  490 WRITE (INTSUB)  ENDATA, ( ZERO,I=1,17 )                             FILGEN
      REWIND INTSUB                                                       FILGEN
      CALL CALGEN                                                         FILGEN
      IF ( ISID .EQ. 0 )  GO TO 900                                       FILGEN
      IMXSEG= 0                                                           FILGEN
      NSEG= 0                                                             FILGEN
      IREGL= 0                                                            FILGEN
      IZONEL = 0                                                          FILGEN
      ISTRL= 0                                                            FILGEN
      ISUBL= 0                                                            FILGEN
      ISEGL = 0                                                           MOD4
      ISKCW= 0                                                            FILGEN
      IF ( ILIST .NE. 0 )  WRITE (IOUT,19)                                FILGEN
   19 FORMAT (20H1  SAMPLE SEGMENT ID/)                                   FILGEN
  500 READ (ISID,3)  ISEG,ICTRY,IREG,IZONE,ISTRAT,ISUB,LAT,LONG           FILGEN
     1  ,ISW,ITRIND,(ITLIST(I),I=1,6),ITYPE                               FILGEN
    3 FORMAT (I4,1X,A4,2I4,2I5,1X,A1,I2,1X,I2,1X,A1,I3,1X,I2,1X,A1        FILGEN
     1  ,1X,I1,1X,6I4,8X,I1)                                              FILGEN
C                                                                         FILGEN
C         TEST FOR END OF SEGMENT ID DATA                                 FILGEN
C                                                                         FILGEN  **-1
  510 IF ( ICTRY .EQ. ENDATA )  GO TO 900                                 FILGEN
      IF(IREG    -  IREGL)997,503,515                                     MOD4
  503 IF(IZONE   -  IZONEL)997,505,515                                    MOD4
  505 IF(ISTRAT  -  ISTRL)997,507,515                                     MOD4
  507 IF(ISUB    -  ISUBL)997,512,515                                     MOD4
  512 IF(ISEG    -  ISEGL)997,997,515                                     MOD4
  515 CONTINUE                                                            MOD4
      NSEGL= NSEGL + 1                                                    FILGEN
      SLAT(2)= LAT(2)                                                     FILGEN
      SLAT(3)= LAT(3)                                                     FILGEN
      SLONG(2)= LONG(2)                                                   FILGEN
      SLONG(3)= LONG(3)                                                   FILGEN
  530 GOOD1= 0                                                            FILGEN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                                                                          FILGEN
C                                                                         FILGEN
C        CHECK DATA ENTRIES ON SAMPLE SEGMENT ID DATA CARD.               FILGEN
C        CONVERT LATITUDE MIN. TO DEGREES AND ADD TO LATITUDE DEGREES.    FILGEN
  600 QLAT=   SLAT(2) + SLAT(3)/60.0                                      FILGEN
      IF ( QLAT .LE. SLATMX )  GO TO 610                                  FILGEN
      CALL ERRMES (3,8)                                                   FILGEN
  610 IF ( LAT(1) .EQ. 1HS )  QLAT= - QLAT                                FILGEN
C        CONVERT LONGITUDE MIN. TO DEGREES AND ADD TO LONGITUDE DEGREES.FILGEN
      QLONG= SLONG(2) + SLONG(3)/60.0                                     FILGEN
      IF ( QLONG .LE. 180.0 )  GO TO 620                                  FILGEN
      CALL ERRMES (3,9)                                                   FILGEN
C        FOR NEGATIVE (WESTERN) LONGITUDES SUBTRACT MAGITUDE FROM 360.    FILGEN
  620 IF ( LONG(1) .EQ. 1HW )  QLONG= 360.0 - QLONG                       FILGEN
C        CONVERT LATITUDE AND LONGITUDE TO RADIANS                        FILGEN
      QLAT= QLAT*DEGTOR                                                   FILGEN
      QLONG= QLONG*DEGTOR                                                 FILGEN
C        CHECK SPRING/WINTER WHEAT INDICATOR.                             FILGEN
      IF ( ISW .NE. 1HS .AND. ISW .NE. 1HW )  CALL ERRMES (3,10)          FILGEN
      IF(ITSEG .EQ. 0)ITRIND = 0                                          FILGEN
      IF(ITSEG .EQ. 0)GO TO 800                                           FILGEN
      IF(ITRIND .EQ. 0)GO TO 650                                          FILGEN
      ICT = 0                                                             FILGEN
      DO 625 I=1,6                                                        FILGEN
      IF(ITLIST(I) .EQ. 0)GO TO 625                                       FILGEN
      ICT = ICT + 1                                                       FILGEN
  625 CONTINUE                                                            FILGEN
      IF(ICT .NE. 0)GO TO 627                                             FILGEN
      CALL ERRMES(3,18)                                                   FILGEN
  627 DO 640 I=1,ICT                                                      FILGEN
      DO 630 J=1,2000                                                     FILGEN
      IF(J .EQ. ILOC1)GO TO 635                                           FILGEN
      IF(ITLIST(I) .EQ. ITRPRT(J))GO TO 640                               FILGEN
  630 CONTINUE                                                            FILGEN
  635 ITRPRT(ILOC1) = ITLIST(I)                                           FILGEN
      ILOC1 = ILOC1 + 1                                                   FILGEN
  640 CONTINUE                                                            FILGEN
      GO TO 800                                                           FILGEN
  650 ITRAIN(ILOC) = ISEG                                                 FILGEN
      ILOC = ILOC + 1                                                     FILGEN
      IF(ILOC .LE. 2000 .AND. ILOC1 .LE. 2000)GO TO 800                   FILGEN
      WRITE(IOUT,18)ILOC,ILOC1                                            FILGEN
      GO TO 1000                                                          FILGEN
C                                                                         FILGEN
  800 IF ( GOODT .NE. 0 )  GO TO 830                                      FILGEN
C        ALL SEGMENT SAMPLE DATA IS GOOD.  WRITE ONE RECORD OF SEGMENT    FILGEN
C        INTERMEDIATE FILE.                                               FILGEN
      IF(FATAL .EQ. 0)NSEGR = NSEGR + 1                                   MOD2
C        COUNT NO. OF SEGMENTS IN THIS SUBSTRATUM.                        FILGEN
      IF ( ISUB .NE. ISUBL )  GO TO 810                                   FILGEN
      IF ( ISTRAT .NE. ISTRL )  GO TO 810                                 FILGEN
      IF ( IZONE .NE. IZONEL )  GO TO 810                                 FILGEN
      IF ( IREG .EQ. IREGL )  GO TO 820                                   FILGEN
C        NEW SUBSTRATUM. ADJUST IMXSEG AND REINITIALIZE COUNTERS.         FILGEN
  810 IMXSEG= MAX0 (IMXSEG,NSEG)                                          FILGEN
      NSEG= 0                                                             FILGEN
      IREGL= IREG                                                         FILGEN
      IZONEL= IZONE                                                       FILGEN
      ISTRL= ISTRAT                                                       FILGEN
      ISUBL= ISUB                                                         FILGEN
  820 NSEG= NSEG + 1
```

```
      ISEGL = ISEG                                                      MOD4
      IF ( FATAL .EQ. 0 )                                               FILGEN
     1  WRITE (INTSEG)  ICTRY,ISEG,IREG,IZONE,ISTRAT,ISUB,QLAT,QLONG    FILGEN
     1  ,ISW,ITRIND,(ITLIST(I),I=1,6)                                   FILGEN
  830 IF ( ILIST .EQ. 0 )  GO TO 500                                    FILGEN
C        IF ILIST .NE. 0, PRINT OUT CARD IMAGES FOR SAMPLE SEGMENT DATA.FILGEN
      WRITE (IOUT,13)  ISEG,ICTRY,IREG,IZONE,ISTRAT,ISUB,LAT,LONG       FILGEN
     1  ,ISW,ITRIND,(ITLIST(I),I=1,6),ITYPE                             FILGEN
   13 FORMAT (1X,I4,1X,A4,2I4,2I5,1X,A1,I2,1X,I2,1X,A1,I3,1X,I2,1X,A1   FILGEN
     1,1X,I1,1X,6I4,8X,I1)                                              FILGEN
      GO TO 500                                                         FILGEN
C                                                                       FILGEN
C        WRITE TRAILER RECORD ON INTERMEDIATE SEGMENT FILE              FILGEN
  900 WRITE (INTSEG)  ENDATA, ( ZERO,I=1,15)                            FILGEN
      REWIND INTSEG                                                     FILGEN
      IMXSEG= MAX0 (IMXSEG,NSEG)                                        FILGEN
      IF ( IMXSEG .LE. MXSEG )  GO TO 990                               FILGEN
      WRITE (IOUT,23)  IMXSEG,MXSEG                                     FILGEN
   23 FORMAT (//40H MAX. NO. OF SEGMENTS IN ANY SUBSTRATA =I5           FILGEN
     1  ,19H  EXCEEDING MXSEG =I5)                                      FILGEN
 1000 STOP                                                              FILGEN
  990 IF(IISEG .EQ. 0)GO TO 999                                         FILGEN
      DO 998 I=1,ILOC1                                                  FILGEN
      ISEGPR = ITMPRI(I)                                                FILGEN
      DO 995 J=1,ILOC                                                   FILGEN
      IF(ITRAIN(J) .EQ. ISEGPR)GO TO 998                                FILGEN
  995 CONTINUE                                                          FILGEN
      WRITE(IOUT,20)ISEGPR                                              FILGEN
   20 FORMAT(//27H TRAINING PRIORITY SEGMENT ,I4,48H IS EITHER NOT A TRAFILGEN
     1INING SEGMENT OR NONEXISTANT)                                     FILGEN
      IERR = IERR + 1                                                   FILGEN
      FATAL = 1                                                         FILGEN
  998 CONTINUE                                                          FILGEN
      GO TO 999                                                         MOD4
  997 FATAL = 1                                                         MOD4
      IERR = IERR + 1                                                   MOD4
      WRITE(IOUT,25)IREG,IZONE,ISTRAT,ISUB,ISEG                         MOD4
   25 FORMAT(1H0,45H SAMPLE SEGMENT ID CARD IS OUT OF ORDER. ID = ,     MOD4
     12I4,3I5)                                                          MOD4
  999 WRITE (IOUT,17)  NSHC,NSSC,NSEGC,NCRCD,NCRECD,NSUBR,NSEGR,        FILGEN
     1NCROP,IERR                                                        FILGEN
   17 FORMAT (//1X,I4,38H  SUBSTRATA HISTORICAL DATA CARDS READ         FILGEN
     1  /1X,I4,39H  SUBSTRATA STATISTICAL DATA CARDS READ               FILGEN
     2  /1X,I4,35H  SAMPLE SEGMENT ID DATA CARDS READ                   FILGEN
     2/1X,I4,31H  CROP CALENDER DATA CARDS READ                         FILGEN
     2/1X,I4,37H  CROP CALENDER ERROR DATA CARDS READ                   FILGEN
     3  /1X,I4,45H  SUBSTRATA INTERMEDIATE FILE RECORDS WRITTEN         FILGEN
     4  /1X,I4,43H  SEGMENT INTERMEDIATE FILE RECORDS WRITTEN           FILGEN
     2/1X,I4,49H  CROP CALENDER INTERMEDIATE FILE RECORDS WRITTEN       FILGEN
     5  ///1X,I4,20H   INPUT ERRORS DETECTED)                           FILGEN
C                                                                       FILGEN
      IF ( FATAL .EQ. 0 )  RETURN                                       FILGEN
      WRITE (IOUT,24)                                                   FILGEN
   24 FORMAT (//36H   FATAL ERROR IN INPUT. JOB ABORTED)                FILGEN
   18 FORMAT(//44H MAX NO OF TRAINING SEGMENTS IS 2000.ILOC = ,I5,9H ILOFILGEN
     1C1 = ,I5)                                                         FILGEN
      STOP                                                              FILGEN
C                                                                       FILGEN
      END                                                               FILGEN
```

```
@ ELT GRDCAL,1,760401, 67418    , 1

      SUBROUTINE GRDCAL(IGRID,JGRID,INDEX,FI,FJ,IFR)                    GRDCAL
C     THIS ROUTINE COMPUTES THE BEST INDEX VALUE FOR A GIVEN GRID PAIR  GRDCAL
C     OF COORDINATES IN THE GLOBAL STEREOGRAPHIC INDEX MATRIX           GRDCAL
      COMMON /CONST/                                                     CONST
     1PI,DEGRAD,      SLATMX,FNDATA,ZERO,INP,IOUT,INISUB,INTSEG,ISIDE,   CONST
     2ICHE,ISHE,LRSIDE,LRCHE,LRSHE,IHIN,MXSEG,ICALIN,ICHE,AREACF         CONST
     3,ISW1,ISW2,IGRDTP                                                  MOD8
      INTEGER ZERO                                                       CONST
      COMMON /DIRAC/                                                     DIRAC
     1ICDIR(252),IDEL(500)                                               DIRAC
      DIMENSION ISTATE(8,8),IBOUND(2,8),IPI(8),ISTAT(64),IBOUN(16)      GRDCAL
      EQUIVALENCE(ISTATE(1,1),ISTAT(1)),(IBOUND(1,1),IBOUN(1))          GRDCAL
      DATA ISTAT/1,3,2,7,8,5,4,6,3,1,2,5,4,7,8,6,5,1,4,5,2,7,6,8,5,3,4,7GRDCAL
     1,6,1,2,8,5,7,6,5,4,1,8,2,7,5,6,1,8,3,4,2,7,1,8,5,6,3,2,4,1,7,8,5,2GRDCAL
     2,5,6,4/                                                           GRDCAL
      DATA IBOUN/1,0,1,1,0,1,-1,1,-1,0,-1,-1,0,-1,1,-1/                 GRDCAL
      CALL RDGRID(JGRID)                                                 GRDCAL
      JGSAV = JGRID                                                      GRDCAL
      IF(IREC(IGRID) .EQ. 0)GO TO 10                                     GRDCAL
      INDEX = IREC(IGRID)                                                GRDCAL
    5 RETURN                                                             GRDCAL
   10 Y = FJ - JGRID                                                     GRDCAL
      X = FI - IGRID                                                     GRDCAL
      IF(Y .NE. 0. .OR. X .NE. 0.)GO TO 15                               MOD9
      ISTA = 1                                                           MOD9
      GO TO 50                                                           MOD9
   15 IF(Y .EQ. 0.)X = X + .0000001                                      MOD9
      SLOPE = Y/X                                                        GRDCAL
      IF(X .GT. 0. .AND. Y .GE. 0.)GO TO 40                              GRDCAL
      IF(X .LT. 0. .AND. Y .GE. 0.)GO TO 30                              GRDCAL
      IF(X .LT. 0. .AND. Y .LE. 0.)GO TO 20                              GRDCAL
      IF(SLOPE .LT. -1.)ISTA = 7                                         MOD9
      IF(SLOPE .GE. -1.)ISTA = 8                                         MOD9
      GO TO 50                                                           GRDCAL
   20 IF(SLOPE .LT. 1.)ISTA = 5                                          MOD9
      IF(SLOPE .GE. 1.)ISTA = 6                                          MOD9
      GO TO 50                                                           GRDCAL
   30 IF(SLOPE .LT. -1.)ISTA = 3                                         MOD9
      IF(SLOPE .GE. -1.)ISTA = 4                                         MOD9
      GO TO 50                                                           GRDCAL
   40 IF(SLOPE .GE. 1.)ISTA = 2                                          MOD9
      IF(SLOPE .LT. 1.)ISTA = 1                                          MOD9
   50 DO 55 I=1,8                                                        GRDCAL
      IPI(I) = ISTATE(I,ISTA)                                            MOD9
   55 CONTINUE                                                           GRDCAL
      DO 75 ICT=1,8                                                      GRDCAL
      IPIT = IPI(ICT)                                                    GRDCAL
      IG = IGRID + IBOUND(1,IPIT)                                        GRDCAL
      JG = JGRID + IBOUND(2,IPIT)                                        GRDCAL
      IF(JG .EQ. JGSAV)GO TO 65                                          GRDCAL
      CALL RDGRID(JG)                                                    GRDCAL
      JGSAV = JG                                                         GRDCAL
   65 IF(IREC(IG) .EQ. 0)GO TO 75                                        GRDCAL
      INDEX = IREC(IG)                                                   GRDCAL
      GO TO 5                                                            GRDCAL
   75 CONTINUE                                                           GRDCAL
   80 IFR = 1                                                            GRDCAL
      GO TO 5
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
* FLT GRID,1,760401, 67416      , 1

               SUBROUTINE GRID(IFRT)                                            GRID
      C        THIS SUBROUTINE CONTROLS THE CALCULATION OF THE INDEX NO         GRID
      C        ASSOCIATED WITH A GIVEN LATITUDE AND LONGITUDE.(FOR LUMP)        GRID
               COMMON /CONST/                                                   CONST
              1PI,DEGTOR,      SLATMX,ENDATA,ZERO,INP,IOUT,INISUB,INISEG,ISIDF, CONST
              2ICWF,ISHF,LRSIDF,LRCWF,LRSHF,IKIN,MXSEG,ICALIN,ICHE,AREALF       CONST
              3,ISW1,ISW2,IGRDTP                                                MODA
               INTEGER ZERO                                                     CONST
               COMMON /INPUT/                                                   INPUT
              1RSEED,SUBAR,HISTPH,DEVTPH,CV(3),DEVTPM,OLAT,OLONG,SLAT(3),SLONG(3)INPUT
              2,NAMSUB(2),ICTRY,ICTRY2,ICASE,ILIST,ITSEG,ISHD,ISSD,ISID,ISCH,   INPUT
              3ISEG,IREG,IZONE,ISTRAT,ISUB,ISW,ITYPE,ITYPE2,INDEX,ISEG2,IREG2,   INPUT
              4IZONE2,ISTRA2,ISUB2,LAT(3),LONG(3),IYR(8),IMO(8),IDA(8),CV4,IGRP, INPUT
              5DELTHF(8),IPLNIF,ISEGSD,ITRIND,NAGR,LAL,ITLIST(6),NCRCD,NCRECD,   INPUT
              6NEROP,IDAYSM(8),DELHF1(8),IPLNI1,ISGSD1,IDAS(8),IMOS(8),IYRS(8)   INPUT
               DOUBLE PRECISION RSEED                                           INPUT
               REAL NAMSUB,ICTRY,ICTRY2                                         INPUT
               DATA I0,J0,PI/125,125,3.1415926/                                 GRID
               IFRT = 0                                                         GRID
               PHIP = ABS(OLAT)                                                 GRID
               X =(PI -2.*PHIP)/4.                                              GRID
               R = 124.8174980*TAN(X)                                           GRID
               A = OLONG - PI/18.                                               GRID
               FI = I0 + R*COS(A)                                               GRID
               FJ = J0 - R*SIN(A)                                               GRID
               IF(OLAT .LT. 0.)FI = FI +2.*I0                                   GRID
               IGRID = FI + .5                                                  GRID
               JGRID = FJ + .5                                                  GRID
               CALL GRDCAL(IGRID,JGRID,INDEX,FI,FJ,IFRT)                        GRID
               IF(IFRT .EQ. 0)GO TO 100                                         GRID
               WRITE(IOUT,900)IREG,IZONE,ISTRAT,ISUB,ISEG                       GRID
          100  RETURN                                                           GRID
          900  FORMAT(1H0, 8H REGION-,I5,6H ZONE-,I5,8H STRATA-,I4,11H SUBSTRATA-GRID
              1,I4,9H SEGMENT-,I4/1H ,73H NO WEATHER GRID INDEX NO. COULD BE FOUNGRID
              2D. THE SEGMENT RECORD IS SKIPPED.)                               GRID
               END                                                              GRID
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
@ ELT IBETAI,1,760405, 55991   , 1
         SUBROUTINE IBETAI(X,A,B,P,IER)                                IBETAI
C                                                                      IBETAI
C     COMPUTE INCOMPLETE BETA INTEGRAL FOR ARGUMENTS                   IBETAI
C     X BETWEEN ZERO AND ONE, A AND B POSITIVE.                        IBETAI
C                                                                      IBETAI
C     X   VALUE TO WHICH FUNCTION IS TO BE INTEGRATED                  IBETAI
C     A   FIRST INPUT PARAMETER                                        IBETAI
C     B   SECOND INPUT PARAMETER                                       IBETAI
C     P   OUTPUT PROBABILITY THAT A RANDOM VARIABLE FROM A             IBETAI
C            BETA DISTRIBUTION HAVING PARAMETERS A AND B               IBETAI
C             WILL BE LESS THAN OR EQUAL TO X                          IBETAI
C    IER  ERROR FLAG WITH POSSIBLE VALUES                              IBETAI
C          =40 X NOT BETWEEN RANGE 0 TO 1                              IBETAI
C          =50  A AND/OR B NOT POSITIVE                                IBETAI
C          =60  GAMMA FUNCTION RANGE VIOLATED (NOT .GT. 0. AND .LE. 88.)  IBETAI
C                                                                      IBETAI
         LOGICAL INDEX                                                 IBETAI
C     TEST FOR ADMISSIBILITY OF ARGUMENTS                              IBETAI
         DATA ACU /1.E-6/                                              IBETAI
         P=X                                                           IBETAI
         IER=50                                                        IBETAI
         IF(A.LE.0.  .OR. B.LE.0.) RETURN                              IBETAI
         IER=40                                                        IBETAI
         IF(X.LE.0.  .OR. X.GE.1.) RETURN                              IBETAI
         IER=0                                                         IBETAI
C                                                                      IBETAI
C     CHANGE TAIL IF NECESSARY AND DETERMINE S                         IBETAI
         ASB=A+B                                                       IBETAI
         CX=1.0-X                                                      IBETAI
         IF(A.GE.ASB*X) GO TO 10                                       IBETAI
         XX=CX                                                         IBETAI
         CX=X                                                          IBETAI
         AA=B                                                          IBETAI
         BB=A                                                          IBETAI
         INDEX=.TRUE.                                                  IBETAI
         GO TO 20                                                      IBETAI
      10 XX=X                                                          IBETAI
         AA=A                                                          IBETAI
         BB=B                                                          IBETAI      *NEW
         INDEX = .FALSE.                                               IBETAI      **-1
      20 TERM=1.0                                                      IBETAI
         AT=1.0                                                        IBETAI
         P=1.0                                                         IBETAI
         NS=BB+CX*ASB                                                  IBETAI
C                                                                      IBETAI
C     USE SOPER REDUCTION FORMULAE                                     IBETAI
         RX=XX/CX                                                      IBETAI
      30 TEMP=BB-AT                                                    IBETAI
         IF(NS.EQ.0) RX=XX                                             IBETAI
      40 TERM=TERM*TEMP*RX/(AA+AT)                                     IBETAI
         P=P+TERM                                                      IBETAI
         TEMP=ABS(TERM)                                                IBETAI
         IF(TEMP.LE.ACU.AND.TEMP.LE.ACU*P) GO TO 50                    IBETAI
         AT=AT+1.0                                                     IBETAI
         NS=NS-1                                                       IBETAI
         IF(NS.GE.0) GO TO 30                                          IBETAI
         TEMP=ASB                                                      IBETAI
         ASB=ASB+1.0
```

```
      GO TO 40                                                          IBETAI
C                                                                       IBETAI
C     CALCULATE RESULT USING GAMMA ROUTINE                              IBETAI
   50 CONTINUE                                                          IBETAI
      CC=AA+BB                                                          IBETAI
      CALL ALGAMA(AA,Y,IER)                                             IBETAI
      IF(IER.GT.0) GO TO 60                                             IBETAI
      CALL ALGAMA(BB,Z,IER)                                             IBETAI
      IF(IER.GT.0) GO TO 60                                             IBETAI
      CALL ALGAMA(CC,W,IER)                                             IBETAI
      IF(IER.GT.0) GO TO 60                                             IBETAI
      BETA=EXP(Y+Z-W)                                                   IBETAI
      P=P*EXP(AA*ALOG(XX)+(BB-1.0)*ALOG(CX))/(AA*BETA)                  IBETAI
      IF(INDEX)P=1.0-P                                                  IBETAI
      RETURN                                                            IBETAI
C     ERROR RETURN                                                      IBETAI
   60 CONTINUE                                                          IBETAI
      P=0.                                                              IBETAI
      RETURN                                                            IBETAI
      END                                                               IBETAI
```

```
 ELT LFPA,1,760401, 67410     , 1

      SUBROUTINE LFPA (FLDA,LMO,LYR,DAYS)                               LFPA
C                                                                       LFPA
C MODULE - P1LFPA                                                       LFPA
C AUGUST 27, 1975                                                       LFPA
C                                                                       LFPA
C                                                                       LFPA
C  JUNE 26,1973                                                         LFPA
C                                                                       LFPA
C  MODULE PPLFPA - POINT TARGET PERFORMANCE PREDICTOR                   LFPA
C                                                                       LFPA
C  GIVEN  DAY,MONTH,YEAR - SUBR. LFPA RETURNS THE RIGHT ASCENSION OF    LFPA
C  GREENWICH AT MIDNIGHT OF A GIVEN DAY                                 LFPA
C                                                                       LFPA
C     REFERENCE EPOCH IS 0 HOUR 1JAN1950                                LFPA
C                                                                       LFPA
C     COMPUTE DAYS IN FULL YEARS FROM EPOCH TO LYR                      LFPA
C     DAYS TO 1JAN1963 IS 4748.                                         LFPA
C                                                                       LFPA
      DAYS=4748.                                                        LFPA
C     NOTE--LYR MUST BE GREATER THAN OR EQUAL TO (19)64                 LFPA
      LASTYR=LYR-1                                                      LFPA
      DO 20 I=63,LASTYR                                                 LFPA
      KREMD=MOD((I-60),4)                                               LFPA
      IF(KREMD.GT.0) GO TO 10                                           LFPA
      DAYS=DAYS+366                                                     LFPA
      GO TO 20                                                          LFPA
   10 DAYS=DAYS+365                                                     LFPA
   20 CONTINUE                                                          LFPA
      IF(LMO-1)30,40,30                                                 LFPA
   30 IF(LMO-2)60,50,60                                                 LFPA
   40 DAYS=DAYS+FLDA-1.                                                 LFPA
      GO TO 270                                                         LFPA
   50 DAYS=DAYS+FLDA+30.                                                LFPA
      GO TO 270                                                         LFPA
   60 KDEL=MOD((LYR-60),4)                                              LFPA
      IF(KDEL.GT.0) GO TO 70                                            LFPA
      DAYS=DAYS+59.                                                     LFPA
      GO TO 80                                                          LFPA
   70 DAYS=DAYS+58.                                                     LFPA
   80 IF(LMO-3) 270,170,90                                              LFPA
   90 IF(LMO-5) 180,190,100                                             LFPA
  100 IF(LMO-7) 200,210,110                                             LFPA
  110 IF(LMO-9) 220,230,120                                             LFPA
  120 IF(LMO-11)240,250,260                                             LFPA
  170 DAYS=DAYS+FLDA                                                    LFPA
      GO TO 270                                                         LFPA
  180 DAYS=DAYS+FLDA+31.                                                LFPA
      GO TO 270                                                         LFPA
  190 DAYS=DAYS+FLDA+61.                                                LFPA
      GO TO 270                                                         LFPA
  200 DAYS=DAYS+FLDA+92.                                                LFPA
      GO TO 270                                                         LFPA
  210 DAYS=DAYS+FLDA+122.                                               LFPA
      GO TO 270                                                         LFPA
  220 DAYS=DAYS+FLDA+153.                                               LFPA
      GO TO 270                                                         LFPA
  230 DAYS=DAYS+FLDA+184.                                               LFPA
      GO TO 270
```

ORIGINAL PAGE IS OF POOR QUALITY

28234-6027-RU-00
Page 242

```
  240 DAYS=DAYS+FLDA+214.                                    LFPA
      GO TO 270                                              LFPA
  250 DAYS=DAYS+FLDA+245.                                    LFPA
      GO TO 270                                              LFPA
  260 DAYS=DAYS+FLDA+275.                                    LFPA
  270 CONTINUE                                               LFPA
C                                                            LFPA
      RETURN                                                 LFPA
C                                                            LFPA
      END                                                    LFPA
```

```
      SUBROUTINE LUMP                                                   LUMP
C        GENERATES OUTPUT FILES                                         LUMP
C          (SEGMENT ID FILE, CROP WINDOW FILE, AND SUBSTRATA HISTORICAL LUMP
C            FILE)                                                      LUMP
C                                                                       LUMP
      COMMON /CONST/                                                    CONST
     1PI,DECTOR,      SLATMX,ENDATA,ZERO,INP,IOUT,INTSUB,INTSEG,ISIDF,  CONST
     2ICWF,ISHF,LRSIDF,LRCWF,LRSHF,IWIN,MXSEG,ICALIN,ICWE,AREACF        CONST
     3,ISW1,ISW2,IGRDTP                                                 MOD8
      INTEGER ZERO                                                      CONST
      COMMON /FLAGS/                                                    FLAGS
     1IERR,IPRNT,GOOD1,NSHC,NSSC,NSEGC,NSUBR,NSEGR,NRSIDF,NRCWF,NRSHF,  FLAGS
     2ISSF,ISTAT,IREGL,IZONEL,ISTRL,ISUBL,ISEGL,NSEG,IMXSEG,NTLIST,     FLAGS
     3READSF,GOOD2,FATAL,IDAYS(8),IDSEG(150)                            FLAGS
     1,IMO1(8),IDA1(8),IYRE(8),IYR1(8),IMO1(8),IDA1(8)                  MOD7
      INTEGER GOOD1,GOOD2,READSF,FATAL                                  FLAGS
      COMMON /INPUT/                                                    INPUT
     1RSEED,SUBAR,HTSTPW,DEVTPW,CV(5),DEVTPM,GLAT,GLONG,SLAT(5),SLONG(5)INPUT
     2,NAMSUB(2),ICTRY,ICTRY2,ICASE,HLIST,ITSEG,ISHF,ISSD,ISID,ISCW,    INPUT
     3ISEG,IREG,IZONE,ISTRAT,ISUB,ISW,ITYPE,ITYPE2,INDEX,ISEG2,IREG2,   INPUT
     4IZONE2,ISTRA2,ISUB2,LAT(3),LONG(3),IYR(8),IMO(8),IDA(8),CV4,ICKP, INPUT
     5DELTWF(8),IPLNTF,ISEGSD,IWIND,NAGR,SAL,ITLIST(6),NCRCD,NCHECK,    INPUT
     6NCROP,IDAYS1(8),DELWF1(8),IPLNT1,ISGSD1,IDAS(8),IMOS(8),IYRS(8)   INPUT
      DOUBLE PRECISION RSEED                                            INPUT
      REAL NAMSUB,ICTRY,ICTRY2                                          INPUT
      COMMON /CALEND/                                                   CALEND
     1IPROBW(8),IPREDB(8),ICRPEW(5),ICRPE3(5)                           CALEND
      COMMON /LDATA/                                                    LDATA
     1WDATA(8),SWDATA(8)                                                LDATA
      INTEGER WDATA,SWDATA                                              LDATA
      REAL ICTRY1                                                       LUMP
C                                                                       LUMP
      DIMENSION  SIDHDR(2), CWHDR(2), SHHDR(2)                          LUMP
      DATA  SIDHDR, CWHDR, SHHDR  / 4HSEGM,3HENT, 8HCROP,4HWIND         LUMP
     1  ,4HSUB ,4HHIST  /,ICV1,IDEVTP,IHISTP/4HCV1 ,6HDEVTP ,6HHISTPH/  LUMP
      DATA IXBAR,ISIGMA/4HXBAR,5HSIGMA/                                 LUMP
C                                                                       LUMP
C                                                                       LUMP
      REWIND ISIDF                                                      LUMP
      REWIND ICWF                                                       LUMP
      REWIND ISHF                                                       LUMP
C        WRITE HEADER RECORDS                                           LUMP
      NFILL= LRSIDF - 3                                                 LUMP
      WRITE(ISIDF)SIDHDR,ICASE,ITSEG,(ZERO,I=1,13)                      MOD5
      NFILL= LRCWF - 4                                                  LUMP
      WRITE (ICWF)  CWHDR,ICASE,IWIN, ( ZERO,I=1,NFILL )                LUMP
      LRSHF = MXSEG +18                                                 MOD6
      NFILL= LRSHF - 4                                                  LUMP
      WRITE (ISHF)  SHHDR,ICASE,MXSEG, ( ZERO,I=1,NFILL )               LUMP
C        INITIALIZE COUNTERS                                            LUMP
      ICTRY2 = 0                                                        LUMP
      READSF = 0                                                        LUMP
      NSEG= MXSEG                                                       LUMP
      ICRPR = 0                                                         LUMP
      ICRPPR = 0                                                        LUMP
      IREGP = 0                                                         LUMP
      IZONP = 0                                                         LUMP
      ISTRP = 0                                                         LUMP
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IGRP1 = 0                                                         MODS
C                                                                       LUMP
C        LOOP TO READ THE SUBSTRATA INTERMEDIATE FILE                   LUMP
  200 READ (INTSUB)  ICTRY,IREG,IZONE,ISTRAT,ISUB,NAHSUB,AREA,HISTPH    LUMP
     1  ,DEVTPH,CV,DEVTPM,NAGP,NAL,CV4,IGRP                             LUMP
C                                                                       LUMP
C        TEST FOR END OF DATA ON SUBSTRATA INTERMEDIATE FILE            LUMP
  203 CONTINUE                                                          LUMP
C                                                                       LUMP
      IF ( ICTRY .NE. ENDATA )  GO TO 205                               LUMP
      ICRPPR = 1                                                        LUMP
  204 IF ( ILIST .NE. 0 )  WRITE (IOUT,11)                              LUMP
   11 FORMAT (//44H  END OF DATA ON SUBSTRATA INTERMEDIATE FILE)        LUMP
C                                                                       LUMP
C        IF EOF ALREADY DETECTED ON SEGMENT INTERMEDIATE FILE,          LUMP
C        THEN TERMINATE OUTPUT FILES AND EXIT.                          LUMP
C        OTHERWISE CONTINUE READING REMAINDER OF SEGMENT INTERMEDIATE   LUMP
C        FILE.                                                          LUMP
      IF ( ICTRY2 .EQ. ENDATA )  GO TO 800                              LUMP
      IF(IGRP1 .EQ. 0)GO TO 100                                         MODS
      IF(IGRP2 .NE. 0)GO TO 100                                         MODS
      WRITE(IOUT,16)IREGP,IZONP,ISTRP                                   MODS
      WRITE(IOUT,19)                                                    MODS
  100 GO TO 290                                                         MODS
C                                                                       LUMP
  205 IF(ICRPPR .NE. 0)GO TO 206                                        MODS
      READ(IGALTH)ICTRY1,IREG1,IZON1,ISTRA1,ISUB1,(IDAYS(I),I=1,8)      LUMP
     1,(DELTHE(I),I=1,4),IPLNTE,ISEGSD,(IDAYS1(I),I=1,8),               LUMP
     2(DELHE1(I),I=1,4),IPLNT1,ISGSO1                                   LUMP
  206 IF(ICTRY1 .EQ. ENDATA)GO TO 211                                   MODS
      IF(IREG - IREG1)211,207,212                                       MODS
  207 IF(IZONE - IZON1)211,208,212                                      LUMP
  208 IF(ISTRAT - ISTRA1)211,209,212                                    LUMP
  209 IF(ISUB - ISUB1)211,230,212                                       LUMP
  230 ICRPR = 0                                                         LUMP
      ICRPPR = 0                                                        LUMP
      GO TO 215                                                         LUMP
  212 ICRPR = 0                                                         MODS
      GO TO 213                                                         LUMP
  211 ICRPR = 1                                                         MODS
  213 ICRPPR = 1                                                        LUMP
  215 ISSUB = 0                                                         LUMP
      IF(IREG .NE. IREGP)GO TO 235                                      LUMP
      IF(IZONE .NE. IZONP)GO TO 235                                     LUMP
      IF(ISTRAT .EQ. ISTRP)GO TO 240                                    LUMP
  235 IF(IGRP1 .EQ. 0)GO TO 237                                         MODS
      IF(IGRP2 .NE. 0)GO TO 237                                         MODS
      WRITE(IOUT,16)IREGP,IZONP,ISTRP                                   MODS
      WRITE(IOUT,19)                                                    MODS
  237 IGRP1 = 0                                                         MODS
      IREGP = IREG                                                      MODS
      IZONP = IZONE                                                     LUMP
      ISTRP = ISTRAT                                                    LUMP
      IGRP2 = 0                                                         LUMP
  240 IF(IGRP .EQ. 2)IGRP1 = 1                                          MODS
      IF(ICRPPR .NE. 0)GO TO 242                                        MODS
      DO 241 I=1,8                                                      MODS
      IPREDA(I) = 0                                                     MODS
      IPREDB(I) = 0                                                     MODS
  241 CONTINUE                                                          MODS
```

```
      DO 243 I=1,5                                                      *NEW
      ICRPEW(I) = 0                                                     *NEW
      ICRPES(I) = 0                                                     *NEW
  243 CONTINUE                                                          *NEW
      CALL CALERR(IDAYS,DELTWE,IPLNTE,IPREDW,ICRPEW)            LUMP    **-2
      CALL CALERR(IDAYS1,DELWE1,IPLNT1,IPREDS,ICRPES)           LUMP
  242 XBAR =HISTPW*(1.0 + DEVTPW)/100.0                         LUMP
      SIGMA = HISTPW*CV(I)/100.0                                MOD1
      IOPT = 0                                                  LUMP
      IER=0                                                     LUMP
      CALL BETAD(RSEED,XBAR,SIGMA,XI,IOPT,IER)                  LUMP
      IF(IER.EQ.0) GO TO 260                                    LUMP
      IF(IER.EQ.1) GO TO 245                                    LUMP
      IF(IER.EQ.2) GO TO 250                                    LUMP
      WRITE(IOUT,900) IREG,IZONE,ISTRAT,ISUB                    LUMP
  900 FORMAT(1H0,8H REGION,I3,7H  ZONE ,I3,8H STRATA ,I4,6H SUBS ,I4,   LUMP
     1/,48H  BETA FUNCTION COULD NOT CONVERGE ON A SOLUTION)    MOD3
      IPWSS=HISTPW                                              LUMP
      GO TO 265                                                 LUMP
  245 IPRNT2=IDEVTP                                             LUMP
      PRNT4 = DEVTPW                                            MOD3
      IPRNT3=IHISTP                                             LUMP-
      PRNT5=HISTPW                                              LUMP
      IPRNT1=IXBAR                                              LUMP
      GO TO 255                                                 LUMP
  250 IPRNT1 = ISIGMA                                           MOD1
      IPRNT2=ICV1                                               LUMP
      IPRNT3=IHISTP                                             LUMP
      PRNT4 = CV(I)                                             MOD1
      PRNT5=HISTPW                                              LUMP
  255 WRITE(IOUT,905)IREG,IZONE,ISTRAT,ISUB,IPRNT1,IPRNT2,IPRNT3,       LUMP
     1PRNT4,PRNT5                                               LUMP
  905 FORMAT(1H0,8H REGION,I3,6H ZONE ,I3,8H STRATA ,I4,6H SUBS ,I4,/,  LUMP
     122H  BETA FUNCTION RESET ,A6,33H VALUE SINCE NOT BETWEEN 0 AND 1.,LUMP
     2/,9H  EITHER ,A6,4H OR ,A6,17H HAVE BAD VALUES ,2F8.3)    MOD3
  260 IPWSS=XI*100.                                             LUMP
  265 IF ( NSEG .EQ. 0 )  GO TO 220                             LUMP
      DO 210 I=1,NSEG                                           LUMP
      IPSEG(I)= 0                                               LUMP
  210 CONTINUE                                                  LUMP
      NSEG= 0                                                   LUMP
C                                                               LUMP
C         IF EOF ALREADY DETECTED ON SEGMENT INTERMEDIATE FILE, LUMP
C         THEN WRITE HISTORICAL FILE AND CONTINUE READING SUBSTRATA     LUMP
C         INTERMEDIATE FILE.                                    LUMP
  220 IF ( ICTRY2 .EQ. ENDATA )  GO TO 450                      LUMP
C         IF READSE .NE. 0, SKIP READING SEGMENT INTERMEDIATE FILE      LUMP
  290 IF ( READSE .NE. 0 )  GO TO 310                           LUMP
C                                                               LUMP
C         LOOP TO READ THE SEGMENT INTERMEDIATE FILE.           LUMP
  300 ISEGCR= ISEGCR + 1                                        LUMP
      READ (INTSEG)  ICTRY2,ISEG,IREG2,IZONE2,ISTRA2,ISUB2,QLAT,QLONG   LUMP
     1  ,ISW,ITRIND,(ITLIST(I),I=1,6)                           LUMP
C                                                               LUMP
C         TEST FOR END OF DATA ON SEGMENT INTERMEDIATE FILE.    LUMP
  303 CONTINUE                                                  LUMP    **-1
C                                                               LUMP
      IF ( ICTRY2 .NE. ENDATA )  GO TO 305                      LUMP
  304 IF ( ILIST .NE. 0 )  WRITE (IOUT,12)                      LUMP
   12 FORMAT (//42H  END OF DATA ON SEGMENT INTERMEDIATE FILE)  LUMP
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C          IF EOF ALREADY DETECTED ON SUBSTRATA INTEMEDIATE FILE,           LUMP
C          THEN TERMINATE OUTPUT FILES AND EXIT.                             LUMP
C          OTHERWISE WRITE ON CROP WINDOW FILE AND HISTORICAL FILE.         LUMP
      IF ( ICTRY .EQ. ENDATA )  GO TO 800                                    LUMP
      GO TO 430                                                              LUMP
C                                                                            LUMP
  305 READSI= 1                                                              LUMP
C          IF EOF ALREADY DETECTED ON SUBSTRATA INTERMEDIATE FILE,          LUMP
C          THEN READ REMAINDER OF SEGMENT INTERMEDIATE FILE WITHOUT         LUMP
C          WRITING ON OUTPUT FILES.                                          LUMP
  310 IF ( ICTRY .EQ. ENDATA )  GO TO 410                                    LUMP
C                                                                            LUMP
C          DOES THE CURRENT SEGMENT BELONG TO THE CURRENT SUBSTRATA RECORDLUMP
      IF ( IREG - IREG2 )  430,312,400                                       LUMP
  312 IF ( IZONE - IZONE2 )  430,314,400                                     LUMP
  314 IF ( ISTRAT - ISTRA2 )  430,316,400                                    LUMP
  316 IF ( ISUB - ISUB2 )  430,318,400                                       LUMP
C          YES.  NOW SEE IF THE CURRENT SEGMENT IS IN THE TRAINING           LUMP
C          INDICATOR LIST.                                                   LUMP
  318 IF(ICRPPR.NE.0) GO TO 415                                              LUMP
      IF(ISW.EQ.1HS)ISWH=1                                                   LUMP
      IF(ISW.EQ.1HW)ISWH=0                                                   LUMP-
      IF(ISWH.EQ.1.AND.IDAYS1(1).EQ.0) GO TO 415                             LUMP
      IF(ISWH.EQ.0.AND.IDAYS(1).EQ.0) GO TO 415                              LUMP
      ISSUB=ISSUB+1                                                          LUMP
  300  NSEG= NSEG + 1                                                        LUMP
       IDSEG(NSEG)= ISEG                                                     LUMP
      CALL GRID(IERT)                                                        MOD8
      IF(IERT .NE. 0)GO TO 300                                               MOD8
C                                                                            LUMP
C          WRITE CURRENT SEGMENT ON SEGMENT ID FILE                          LUMP
      WRITE(ISIDF)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISEG,ITRIND,(ITLIST(I),      MOD3
     1I=1,6),QLAT,QLONG,INDEX,ISWH                                           MOD3
C*****                                                                       LUMP
      IF ( HIST .EQ. 2 )                                                     LUMP
     1WRITE(IOUT,21)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISEG,ITRIND,(ITLIST(I)     MOD3
     2,I=1,6),QLAT,QLONG,INDEX,ISWH                                          MOD3
   21 FORMAT(1H0,17H  SEGMENT ID FILE/,                                                *NEW
     19H   ICTRY =,A4,9H   IREG =,I4,10H   IZONE =,I4,                                  *NEW
     210H   ISTRAT =,                                                                   *NEW
     1I5,9H   ISUB =,I5,9H   ISEG =,I5,11H   ITRIND =,I2/                      MOD3     **-2
     211H   ITLIST =,6I5/9H   QLAT =,E14.6,10H   QLONG =,E14.6,                MOD3
     310H   INDEX =,I5,9H   ISWH =,I2)                                         MOD3
      NRSIDF= NRSIDF + 1                                                     LUMP
      READSF= 0.                                                             LUMP
      GO TO 300                                                              LUMP
C                                                                            LUMP
C          INCONSISTENCY BETWEEN SUBSTRATA DATA AND SEGMENT DATA.            LUMP
  400 WRITE (IOUT,13)  ISEG,ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2                 LUMP
     1  ,ICTRY,IREG,IZONE,ISTRAT,ISUB                                        LUMP
   13 FORMAT(//53H THE FOLLOWING SEGMENT DATA WAS DROPPED DUE TO DISAGR LUMP
     1 31HEEMENT WITH THE SUBSTRATA DATA./10H   SEGMENT,I5,11H   COUNTRYLUMP
     2 ,A4,9H   REGION,I4/1H   ZONE,I4,10H   STRATUM,I5,13H   SUBSTRATUMLUMP
     3 ,I5,19H  FROM SEGMENT DATA/15X,11H   COUNTRY ,A4,9H   REGION,I4  LUMP
     4 ,7H   ZONE,I4,10H   STRATUM,I5,13H   SUBSTRATUM,I5,21H  FROM SUBSLUMP
     5TRATA DATA)                                                            LUMP
      GO TO 300                                                              LUMP
C                                                                            LUMP
  410 WRITE (IOUT,14)  ISEG,ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2                 LUMP
   14 FORMAT (//51H THE FOLLOWING SEGMENT DATA WAS DROPPED BECAUSE IT       LUMP
```

```
     1  63HDID NOT BELONG TO ANY STRATA ON THE SUBSTRATA HISTORICAL FILELUMP
     2  /10H   SEGMENT,I5,11H   COUNTRY ,A4,9H   REGION,I4,7H   ZONE,I4 LUMP
     3  ,10H   STRATUM,I5,13H   SUBSTRATUM,I5)                          LUMP
      GO TO 300                                                         LUMP
  415 WRITE(IOUT,15)IREG,IZONE,ISTRAT,ISUB,ISEG                         LUMP
   15 FORMAT(1H0,8H REGION ,I3,6H ZONE ,I3,8H STRATA ,I4,11H SUBSTRATA  LUMP
     1,I4,9H SEGMENT ,I4,/,93H NO CROP CALENDER DATA AVAILABLE FOR SEGMEMOD4
     2NT INTERMEDIATE FILE RECORD. SEGMENT RECORD SKIPPED )            MOD3
      GO TO 300                                                         LUMP
C                                                                       LUMP
C         WRITE RECORD ON CROP WINDOW FILE                              LUMP
  430 CONTINUE                                                          MOD3
      IF(IGRP .EQ. 1)GO TO 440                                          LUMP
      IF(IGRP.EQ.2) GO TO 445                                           LUMP
      IF(ISSUB.EQ.0)GO TO 448                                           LUMP
      FATAL=1                                                           LUMP
      WRITE(IOUT,16)IREG,IZONE,ISTRAT,ISUB                              LUMP
   16 FORMAT(1H0,8H REGION ,I3,6H ZONE ,I3,8H STRATA ,I4,11H SUBSTRATA  LUMP
     1,I4)                                                              LUMP
      WRITE(IOUT,17)                                                    LUMP
   17 FORMAT(51H  A GROUP 3 SUBSTRATA HAS 1 OR MORE SAMPLE SEGMENTS)    LUMP
      GO TO 448                                                         LUMP
  440 IF(ISSUB.NE.0) GO TO 448                                          LUMP
      IGRP=3                                                            LUMP
      WRITE(IOUT,16)IREG,IZONE,ISTRAT,ISUB                              LUMP
      WRITE(IOUT,18)                                                    LUMP
   18 FORMAT(61H  A GROUP 1 SUBSTRATA DOES NOT HAVE AT LEAST 1 SAMPLE SELUMP
     1GMENT)                                                            LUMP
      GO TO 448                                                         LUMP
  445 IF(IREG.NE.IREG2) GO TO 447                                       LUMP
      IF(IZONE .NE. IZONE2) GO TO 447                                   LUMP
      IF(ISTRAT .NE. ISTRA2) GO TO 447                                  LUMP
      IGRP2= IGRP2+ ISSUB                                               LUMP
      GO TO 448                                                         LUMP
  447 CONTINUE                                                          MOD5
   19 FORMAT(49H  A STRATA WITH AT LEAST 1 GROUP 2 SUBSTRATA DOES,      MOD1
     151H NOT HAVE A SAMPLE SEGMENT IN ANY GROUP 2 SUBSTRATA)           MOD1
  448 IF(ICRPPR .NE. 0)GO TO 450                                        MOD3
      NRCWF = NRCWF + 1                                                 MOD3
      WRITE (ICWF)  ICTRY,IREG,IZONE,ISTRAT,ISUB, (IPREDW(I),I=1,8),    LUMP
     1ISEGSD,(ICRPEW(I),I=1,5),(IPREDS(I),I=1,8),ISGSDI,(ICRPES(I),I=1,5LUMP
     2)                                                                 LUMP
C*****                                                                  LUMP
      IF ( ILIST .EQ. 2 )                                               LUMP
     1WRITE(IOUT,24)ICTRY,IREG,IZONE,ISTRAT,ISUB,IPREDW,ISEGSD,         MOD3
     2ICRPEW,IPREDS,ISGSDI,ICRPES                                       LUMP
   24 FORMAT(1H0,18H  CROP WINDOW FILE/10H   ICTRY =,A4,9H   IREG =,I4,          *NEW
     110H   IZONE =,I4,11H   ISTRAT =,I5,9H   ISUB =,I5/                         *NEW
     210H IPREDW = ,8I6/10H ISEGSD = ,I2,10H ICRPEW = ,5I3/,          LUMP-  **-2
     310H IPREDS = ,8I6/,10H ISGSDI = ,I2,10H ICRPES = ,5I3/)           LUMP
C         WRITE RECORD ON SUBSTRATA HISTORICAL FILE                     LUMP
  450 NRSHF = NRSHF + 1                                                 LUMP
      WRITE (ISHF)  ICTRY,IREG,IZONE,ISTRAT,ISUB,NSEG,( IDSEG(I),I=1,   LUMP
     1  MXSEG ),IGRP,    HISTPW,AREA,IPWSS,NAGR,NAT,DEVTPW,DEVTPM,CV,CV4 LUMP
C*****                                                                  LUMP
      IF ( ILIST .NE. 2 )  GO TO 200                                    LUMP
      WRITE(IOUT,22)ICTRY,IREG,IZONE,ISTRAT,ISUB,NSEG,(IDSEG(I),I=1,    MOD3
     1       NSEG )                                                     LUMP
   22 FORMAT(1H0,27H  SUBSTRATA HISTORICAL FILE/10H   ICTRY =,A4,                *NEW
     19H   IREG =,I4,10H   IZONE =,I4,11H   ISTRAT =,I5,9H   ISUB =,I5,          *NEW
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      29H   NSFG =,I4/10H   TDSEG =,16I6/(10X,16I6))                    *NEW
       WRITE(IOUT,23)IGRP,HISTPW,AREA,TPKSS,NAGR,NAL,DEVTPW,DEVTPM,   MOD3  **-3
      1CV,CV4                                                         LUMP
   23 FORMAT(11H    IGRP  =,I2,11H   HISTPW =,E14.6,9H   AREA =,E14.6,      *NEW
      110H   TPKSS =,E14.6/10H NAGR   = ,I5,10H NAL    = ,I5,               *NEW
      110H DEVTPW = ,F6.3,10H DEVTPM = ,F6.3,/10H CV     = ,4(F5.3,1X))  MOD4  **-2
       GO TO 200                                                      LUMP
C                                                                     LUMP
C         WRITE TERMINAL RECORDS ON OUTPUT FILES.                     LUMP
  800  NFILL= LRSIDF - 1                                              LUMP
       WRITE (ISIDF)  ENDATA, ( ZERO,I=1,NFILL )                      LUMP
       REWIND ISIDF                                                   LUMP
       NFILL= LRCWF - 1                                               LUMP
       WRITE ( ICWF)  ENDATA, ( ZERO,I=1,NFILL )                      LUMP
       REWIND ICWF                                                    LUMP
       NFILL= LRSWF - 1                                               LUMP
       WRITE (ISWF)  ENDATA, ( ZFRO,I=1,NFILL )                       LUMP
       REWIND ISWF                                                    LUMP
       RETURN                                                         LUMP
       END                                                            LUMP
```

```
@ FLT LUMPDR,1,760405, 55992      , 1

C      PROGRAM LUMPDR(ITEST=401,OTEST=401,TAPE5=ITEST,TAPE6=OTEST        MOD1
C     1,SHISTD=401,SSTATD=401,SEGIDD=401,SEGCWD=401,SEGID=604,CHOPW=604,  MOD8
C     2SUBHST=604,CRPERR=401,CRPINT=604,TAPE30=504,TAPE31=504,           MOD8
C     3INDMAT=1004,TAPE4=INDMAT                                          MOD8
C     2   ,TAPE11=SHISTD,TAPE12=SSTATD,TAPE13=SEGIDD,TAPE14=SEGCWD        LUMPDR
C     4,TAPE1=SEGID,TAPE2=CHOPW,TAPE3=SUBHST,                            MOD8
C     4TAPE56=CRPERR,TAPE57=CRPINT)                                      LUMPDR
       COMMON /CONST/                                                    CONST
      1PI,DEGTOR,          SLATMX,ENDATA,ZERO,INP,IOUT,INTSUB,INTSEG,ISIDF, CONST
      2ICWF,ISHF,LRSIDF,LRCWF,LRSHF,IWIN,MXSEG,ICALIN,ICWE,AREACF         CONST
      3,ISW1,ISW2,IGRDTP                                                 MOD8
       INTEGER ZERO                                                      CONST
       COMMON /FLAGS/                                                    FLAGS
      1IERR,IPRNT,GOOD1,NSHC,NSSC,NSEGC,NSUBR,NSEGR,NRSIDF,NRCWF,NRSHF,   FLAGS
      2ISSC,ISTAT,IREGL,IZONEL,ISTRL,ISUBL,ISEGL,NSEG,IMXSEG,NTLIST,      FLAGS
      3READSF,GOOD2,FATAL,IDAYS(8),IPSF,(150)                            FLAGS
      1,IMO1(8),IDA1(8),IYR1(8),IYR1(8),IMO1(8),IDA1(8)                  MOD2
       INTEGER GOOD1,GOOD2,READSF,FATAL                                  FLAGS
       COMMON /INPUT/                                                    INPUT
      1RSEED,SUBAR,HISTPW,DEVTPW,CV(3),DEVTPM,OLAT,OLONG,SLAT(3),SLONG(3)INPUT
      2,NAMSUB(2),ICTRY,ICTRY2,ICASE,ILIST,ITSEG,ISHD,ISSD,ISID,ISCH,     INPUT
      3ISEG,IREG,IZONE,ISTRAT,ISUB,ISW,ITYPE,ITYPE2,INDEX,ISEG2,IREG2,    INPUT
      4IZONE2,ISTRA2,ISUB2,LAT(3),LONG(3),IYR(8),IMO(8),IDA(8),CV4,IGRP,  INPUT
      5DELTWE(4),IPLNTF,ISEGSD,ITRIND,NAGR,NAL,ITLIST(6),NCRCD,NCRFCD,    INPUT
      6NCKOP,IDAYS1(8),DELWE1(4),IPLNT1,ISGSD1,IDAS(8),IMOS(8),IYRS(8)    INPUT
       DOUBLE PRECISION RSEED                                            INPUT
       REAL NAMSUB,ICTRY,ICTRY2                                          INPUT
       COMMON /LDATA/                                                    LDATA
      1WWDATA(8),SWDATA(8)                                               LDATA
       INTEGER WWDATA,SWDATA                                             LDATA
C                                                                        LUMPDR
       DATA  PI          , DEGTOR       , AREACF      , SLATMX , ENDATA   LUMPDR
      1    / 3.1415926 , 0.017453292 , 4.046873E-5, 65.0    , 4HZZZZ  /  LUMPDR
       DATA  ZERO  , INP , IOUT , INTSUB , INTSEG , ISIDF , ICWF , ISHF   LUMPDR
      1    /   0   ,  5  ,  6   ,   13   ,   14   ,   1   ,  2   ,  3  /  LUMPDR
       DATA  LRSIDF , LRCWF , LRSHF , IWIN , MXSEG , ICWF ,ICALIN         LUMPDR
      1    /   17   ,  35   ,  170  ,  4   ,  150  ,  8   ,  7  /         MOD1
C                                                                        LUMPDR
       DATA FATAL/0/,IGRDTP/4/                                           MOD9
C                                                                        LUMPDR
C                                                                        LUMPDR
C*****        TEMPORARILY REWIND INPUT FILES *****                       LUMPDR
C*****                                                                   LUMPDR
C******                                                                  LUMPDR
       REWIND INTSUB                                                     LUMPDR
       REWIND INTSEG                                                     LUMPDR
       REWIND ICALIN                                                     LUMPDR
C                                                                        LUMPDR
C          READ CONTROL CARD                                             LUMPDR
       READ (INP,1)  ICASE,RSEED,ILIST,ITSEG,ISHD,ISSD,ISID,ISCH,ICWE     LUMPDR
     1 FORMAT (I5,5X,D12.0,I7,6I5)                                       LUMPDR
       WRITE (IOUT,10)  ICASE,RSEED,ILIST,ITSEG,ISHD,ISSD,ISID,ISCH,ICWE  LUMPDR
    10 FORMAT (1H1  CASE NO.,I34                                         LUMPDR
      1  /2X10  INITIAL RANDOM NUMBER SEED,D19.12                        LUMPDR
      2  /1X10  LIST OPTION,I31                                          LUMPDR
      3  /2X10  TRAINING SEGMENT OPTION,I19                              LUMPDR
      4  /3X10  SUBSTRATA HISTORICAL DATA FILE NO.,I8                    LUMPDR
      5  /3X10  SUBSTRATA STATISTICAL DATA FILE NO.,I7                   LUMPDR
```

```
      6  /34H0  SAMPLE SEGMENT ID DATA FILE NO.,I11                    LUMPDR
      7  /35H0  SUBSTRATA CROP CALENDER FILE NO.,5X,I5                  MOD2
      8  /36H0  CROP CALENDER ERROR DATA FILE NO.,4X,I5)                MOD2
C                                                                        LUMPDR
       REWIND ISMD                                                       LUMPDR
       REWIND ISSD                                                       LUMPDR
       REWIND ISTD                                                       LUMPDR
       REWIND ISCW                                                       LUMPDR
       REWIND ICFE                                                       LUMPDR
       CALL RDGRID(0)                                                    MOD8
C                                                                        LUMPDR
       IF ( ILIST .NE. 0 )  WRITE (IOUT,11)  ICASE,RSEED,ILIST,ITSFG,ISMDLUMPDR
      1  ,ISSD,ISTD,ISCW,ICWF                                            LUMPDR
   11  FORMAT (30H1 LISTING OF INPUT CARD IMAGES///16H    CONTROL CARD// LUMPDR
      11X,I5,3X,D19.12,I7,6I5)                                                    *NEW
C                                                                        LUMPDR   **-1
   50  IF ( ISMD .NE. 0 .OR. ISTD .NE. 0 )  CALL FILGEN                  LUMPDR
       CALL LUMP                                                         LUMPDR
       WRITE(IOUT,905)NRSIDF,NRCWF,NRSHF                                 MOD3
  905 FORMAT(1H0,I4,32H SEGMENT ID FILE RECORDS WRITTEN/                 MOD1
      1,1X,I4,35H CROP CALENDER FILE RECORDS WRITTEN/                    MOD1
      2,1X,I4,39H SEGMENT REFERENCE FILE RECORDS WRITTEN)                MOD1-
       IF(FATAL .NE. 0)WRITE(IOUT,900)                                   LUMPDR
  900 FORMAT(1H0,55H FATAL ERROR OCCURED IN LUMP, FILES MAY NOT BE ANY GLUMPDR
      1OOD)                                                              LUMPDR
       WRITE (IOUT,13)  RSEED                                            LUMPDR
   13  FORMAT(17H1  JOB COMPLETED.//28H   LAST RANDOM NUMBER SEED =      LUMPDR
      1D19.12)                                                           LUMPDR
       CALL RDGRID(-1)                                                   MOD8
       STOP                                                              LUMPDR
       END                                                               LUMPDR
```

```
• ELT RDGRID,1,760401, 67419    , 1

         SUBROUTINE RDGRID(JG)                                          RDGRID
C        THIS ROUTINE WILL OPEN OR CLOSE IGRDTP FILE OR WRITE A RECORD ON IRDGRID
C        T                                                              RDGRID
         COMMON /CONST/                                                 CONST
        1PI,DEGIOR,      SLATMX,FNDATA,ZFRO,INP,IOUT,INTSUB,INTSEG,ISIDF,  CONST
        2ICHF,ISHF,LRSIDF,LRCHF,LRSHF,IWIN,MXSEG,ICALIN,ICWL,ARLAUF     CONST
        3,ISW1,ISW2,IGRDTP                                              MODA
         INTEGER ZERO                                                   CONST
         COMMON /DIRAC/                                                 DIRAC
        1IFDIN(252),IREC(500)                                           DIRAC
         DATA MAXJ/42/,IMAT1/4HINDE/,IMAT2/4HXMAT/                      RDGRID
C        DATA MAXJ/252/                                                 RDGRID
         IF(JG)25,10,40                                                 RDGRID
      10 DEFINE FILE 4(42,500,U,IDUM)
         READ(IGRDTP'1,ERR=100)(IREC(I),I=1,500)
         IF(IREC(1) .EQ. IMAT1 .AND. IREC(2) .EQ. IMAT2)GO TO 50        RDGRID
         WRITE(IOUT,900)                                                RDGRID
         STOP                                                           RDGRID
     900 FORMAT(1H0,85H *** WRONG FILE MOUNTED FOR INDEX GRID MATRIX)   RDGRID
      25 CONTINUE                                                       -
C     25 CALL CLOSE(IGRDTP,0)
         GO TO 50                                                       RDGRID
      40 J = JG + 1                                                     RDGRID
         READ(IGRDTP'J,ERR=100)(IREC(I),I=1,500)
      50 RETURN                                                         RDGRID
     100 WRITE(IOUT,901)
         STOP
     901 FORMAT(1H0,91H *** AN IRRECOVERABLE I/O ERROR HAS OCCURRED ON READ
        1ING A RECORD FROM THE INDEX MATRIX FILE /10X,31H THE JOB IS BEING
        2ABANDONED ***)
         END                                                            RDGRID
```

```
@ FLT RDM1A,1,760401, 67411      , 1

      SUBROUTINE RDM1A(FL,U)                                            RDM1A
C     RAMDUM NUMBER GENERATOR FOR ALL COMPUTERS                         RDM1A
      DOUBLE PRECISION C1,C2,R1,R2,T,FL,TH035,ONE,ZERO,XMOD,YMOD        RDM1A
      DATA C1,C2,ONE,ZERO/3051749376.D1,84365.D0,1.D0,0.D0/,            RDM1A
     1     TH035/34359738368.D0/                                        RDM1A
      XMOD(T) = DMOD(T,TH035)                                           RDM1A
      YMOD(T) = DMOD(T,262144.D0)                                       RDM1A
      T = FL                                                            RDM1A
      IF(T .LE. ZERO) T = ONE                                           RDM1A
      R2 = YMOD(T)                                                      RDM1A
      R1 = T - R2                                                       RDM1A
      T = XMOD(XMOD(C1*R2+C2*R1) + C2*R2)                               RDM1A
      U = T/TH035                                                       RDM1A
      FL = T                                                            RDM1A
      RETURN                                                            RDM1A
      END                                                               RDM1A

 11. TOC
```

ORIGINAL PAGE IS OF POOR QUALITY

SEE BOOK IV

Table of Contents

PART I

PROBLEM DESCRIPTION

Problem Description for System Error Executive (SEE) Program

1.0 SCOPE

1.1 PROGRAM CAPABILITIES

This program provides the means to create certain files containing error coefficients and terms necessary as input to the LEM program. The input data to this program consists of three card sets containing YES error and truth data, CAMS error data and signature extension data. As the data on the cards is transferred from card images to the output data files, each data field is checked for validity and each card ID group is checked against the substrata historical file to detect missing data. A message is printed for each error found. The program provides on option a list of all cards and also a listing of all records written on the output files. The output files are the YES error file, CAMS error file and signature extension file.

1.2 PROGRAM DEVELOPMENT AND ORGANIZATION

The program will be developed in FORTRAN IV on the CDC 6600 time sharing system and later converted to FORTRAN V in order to be run on the UNIVAC 1110 at NASA, Houston, under EXEC VIII.

1.3 OPERATIONAL ASSUMPTIONS

- One case may be run at one time.
- Only one country may be processed in a case.
- The three data card sets must be on separate input files and in sort as follows:
  The major sort field is listed first.

  1. YES error data, set 5

     Country, region, zone, strata and card sequence number

  2. CAMS error data set 6

     Country, region, zone, strata, substrata, segment and card sequence number

3. Signature extension set 7

Country, region, zone and card sequence number

- All cards in a group with the same ID must be present even though a card may contain only blank data entries.
- The program will require less than 20,000 words decimal of storage in the CPU of the UNIVAC.
- The program will run only if the specified substrata historical file is present.
- Except in the case of an input card set being out of sort, the program will continue checking for field errors and for mismatches vs. the substrata historical file. A fatal error will stop output files from being written.
- Input data groups for which there is no ID match on the substrata historical file are considered to be extra data and are not written out on the output files.
- IDFRS and IDTOS specify ID limits for which certain errors are considered to be fatal. If a normally fatal error occurs on a card group outside these limits, it will be considered to be non-fatal.
- A negative non-zero entry in any card input unit designator IYESR, ISIGEX or ICAMER indicates that that data is not to be read or processed and the corresponding output file will not be created.
- The United States must be assigned the symbol "USA " and Canada must be assigned the symbol "CAND".

2.0 INPUT

There are four card sets and one binary data file required as input to this program.

The card sets are:

1. Program control card
2. YES truth and error data
3. CAMS error data
4. Signature extension data

The data file is:

1. Substrata historical file

2.1 CARD INPUT

2.1.1 List of Data Quantities

See input data description sheets on Pages 259-263.

2.1.2 Card Formats

See card format sheets on Pages 264-267.

2.1.3 Deck Setup

The input card data stream will have one card, the program control card.

The YES data will be on the file specified in IYESR (Unit 1 is assumed if this entry is 0).

The CAMS error data will be on the file specified in ICAMER (Unit 4 is assumed if this entry is 0).

The signature extension data will be on the file specified in ISIGEX (Unit 2 is assumed if this entry is 0).

In the case of the signature extension and CAMS error data cards, if the data is constant for a particular ID level such as zone, only one card group may be filled out for that zone and all ID levels below zone are left blank.

INPUT DATA DESCRIPTION

Name	Symbol	Dimension	Nominal Value	Range	Units	Description
CASE	-	1	-	1-9999	-	Case no. to be assigned to identify the three output files.
LIST	-	1	-	0-2	-	List option to list all input data cards or only those in error and all output file records. = 0 list cards in error = 1 list all input data cards = 2 list all input data cards and all output file records
DFRS	-	2	0, 0		-	Specifies ID of starting zone for which errors are to be considered fatal. 1st item region, 2nd - zone. Blank or 0 entry means entire ID range is active.
DTOS	-	2	0, 0		-	Specifies ID of ending zone for which errors are considered to be fatal. 1st item is reg., 2nd - zone.
YESU	-	1	1	-	-	File unit no. for YES input card data. Minus value means not to read or process this data.
SIGEX	-	1	2	-	-	File unit no. for signature extension input card data. Minus value means not to read or process this data.
CAMER	-	1	4	-	-	File unit no. for CAMS error input card data. Minus value means not to read or process.
CSESH	-	1	-	-	-	Zero or blank means not to check on case no. of input substrata historical file.
CTRY	-	1	-	-	-	Four character country name (i.e., USA , USSR).

DESCRIPTION

Units	Description
-	Region no.
-	Zone no.
-	Strata no.
-	Substrata no.
-	Segment no.
Quin/ hectar	True yield of stratum (for USA , and CAND - bushels/acre.
Yr, Mon, Day	Year, month and day for error truncation. If 1st value is 0, then there is no bias or std. dev. data. Any other 0 value terminates data entries.
Quin/ hectar	Bias error of yield. One value for each ≠0 value of IERDTE for USA , CAND - bushels/ acre.
Quin/ hectar	Standard deviation of yield error. One value for each ≠0 value of IERDTE for USA , CAND - bushels/acre.
-	Nominal frequency of labeling wheat given pure wheat pixels (conditional prob.). One for each window.
-	Nominal frequency of labeling wheat given mixed pixels (cond. prob.). One for each window.
-	Nominal frequency of labeling wheat given pure other pixels (cond. prob.). One for each window.

INPUT DATA DESCRIPTION

Name	Symbol	Dimension	Nominal Value	Range	Units	Description
BWW	$B(W/W)_i$	4	0	± 9.999	-	Bias error in classifying pure wheat pixels. One for each window.
BWM	$B(W/M)_i$	4	0	± 9.999	-	Bias error in classifying mixed pixels. One for each window.
BWO	$B(W/O)_i$	4	0	± 9.999	-	Bias error in classifying other crop pixels. One for each window.
SIGWW	$\sigma(W/W)_i$	4	0	0-9.99	-	Standard deviation in classifying wheat pixels. One for each window.
SIGWM	$\sigma(W/M)_i$	4	0	0-9.99	-	Standard deviation in classifying mixed pixels. One for each window.
SIGWO	$\sigma(W/O)_i$	4	0	0-9.99	-	Standard deviation in classifying other crop pixels. One for each window.
BPW	B_{PW_i}	4	0	± 9.999	-	Bias in proportion estimate (model #2). One for each window.
SIGPW	σ_{PW_i}	4	0	0-9.99	-	Standard deviation in proportion estimate (model #2).
			FOR MODEL 1 SIGNATURE EXTENSION			
B1W	B_{1W}	1	0	± 9.999	-	Multiplicative bias error W/W.
B2W	B_{2W}	1	0	± 9.999	-	Additive bias error W/W.
SIG1W	σ_{1iW}	6	-	0-9.99	-	Multiplicative std. dev. error (W/W). One for each training priority segment.
SIG2W	σ_{2iW}	6	-	0-9.99	-	Additive std. dev. error (W/W). One for each training priority segment.

INPUT DATA DESCRIPTION

Name	Symbol	Dimension	Nominal Value	Range	Units	Description
B1M	B_{1M}	1	-	± 9.999	-	Multiplicative bias error (W/M).
B2M	B_{2M}	1	-	± 9.999	-	Additive bias error (W/M).
SIG1M	σ_{1iM}	6	-	0-9.99	-	Multiplicative std. dev. error (W/M). One for each training priority segment.
SIG2M	σ_{2iM}	6	-	0-9.99	-	Additive std. dev. error (W/M). One for each training priority segment.
B1O	B_{1O}	1	-	± 9.999	-	Multiplicative bias error (W/O).
B2O	B_{2O}	1	-	± 9.999	-	Additive bias error (W/O).
SIG1O	σ_{1iO}	6	-	0-9.99	-	Multiplicative std. dev. error (W/O). One for each training priority segment.
SIG2O	σ_{2iO}	6	-	0-9.99	-	Additive std. dev. error (W/O). One for each training priority segment.
			FOR MODEL 2 SIGNATURE EXTENSION			
B1	B_1	1	-	± 9.999	-	Multiplicative bias error
B2	B_2	1	-	± 9.999	-	Additive bias error
SIG1	σ_{1i}	6	-	0-9.99	-	Multiplicative std. dev. error. One for each training priority segment.
SIG2	σ_{2i}	6	-	0-9.99	-	Additive std. dev. error. One for each training priority segment.
ISET	-	1	-	-	-	Card set number in column 80. 5 for YES error data. 6 for CAMS error data. 7 for signature extension data.

INPUT DATA DESCRIPTION

Name	Symbol	Dimension	Nominal Value	Range	Units	Description
SEQ	-	1	-	-	-	Card sequence number within card set with the same ID. Used to identify each card of a group. YES error has a 2 card group. CAMS error and sig. ext. each have 4 card groups.

SEE Control Card

1	5	6	9	12	15	18	20	22	24
ICASE	ILIST	IDFRS		IDTOS		IYESU	ISIGEX	ICAMER	ICSESH
		Region	Zone	Region	Zone				
I4	I1	I3	I3	I3	I3	I2	I2	I2	I4

YES Error Data Card 1

1	5	9	12	15	19	24	30	36	42	48	54	60	79	80
X	Country	Reg	Zone	Strata	Y	Error Trunc. Date 1	Date 2	Date 3	Date 4	Date 5	Date 6			
						Y M D	Y M D	Y M D	Y M D	Y M D	Y M D		1	5
	A4	I3	I3	I4	F5.2	3I2	3I2	3I2	3I2	3I2	3I2			

YES Error Data Card 2

1	5	9	12	15	19	24	29	34	39	44	49	54	59	64	69	74	79	80
X	Country	Reg	Zone	Strata	Bias Error of Yield						Standard Error of Yield							
					B_1	B_2	B_3	B_4	B_5	B_6	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	2	5
	A4	I3	I3	I4	6F5.1						6F5.2							

SEE Data Cards

CAMS Error Data Card 1 for Pure Wheat Pixels

	5	9	12	15	19	23	27	31	35	39	45	51	57	63	67	71	75	79	80
						Nominal Freq. Label Wht.				Bias Error in Classifying Wheat				Std. Dev. Classify Wht.					
-nt	Country	Reg.	Zone	Strata	Sub-strata	$P(W/W)_1$	$P(W/W)_2$	$P(W/W)_3$	$P(W/W)_4$	$B(W/W)_1$	$B(W/W)_2$	$B(W/W)_3$	$B(W/W)_4$	$\sigma(W/W)_1$	$\sigma(W/W)_2$	$\sigma(W/W)_3$	$\sigma(W/W)_4$	1	6
	A4	I3	I3	I4	I4	4F4.2				4F6.3				4F4.2					

CAMS Error Data Card 2 for Mixed Pixels

	5	9	12	15	19	23	27	31	35	39	45	51	57	63	67	71	75	79	80
						Nominal Freq. Label Wht.				Bias Error in Classifying Wheat				Std. Dev. Classify Wht.					
-nt	Country	Reg.	Zone	Strata	Sub-strata	$P(W/M)_1$	$P(W/M)_2$	$P(W/M)_3$	$P(W/M)_4$	$B(W/M)_1$	$B(W/M)_2$	$B(W/M)_3$	$B(W/M)_4$	$\sigma(W/M)_1$	$\sigma(W/M)_2$	$\sigma(W/M)_3$	$\sigma(W/M)_4$	2	6
	A4	I3	I3	I4	I4	4F4.2				4F6.3				4F4.2					

CAMS Error Data Card 3 for Other Pixels

	5	9	12	15	19	23	27	31	35	39	45	51	57	63	67	71	75	79	80
						Nominal Freq. Label Wht.				Bias Error in Classifying Wheat				Std. Dev. Classify Wht.					
-nt	Country	Reg.	Zone	Strata	Sub-strata	$P(W/O)_1$	$P(W/O)_2$	$P(W/O)_3$	$P(W/O)_4$	$B(W/O)_1$	$B(W/O)_2$	$B(W/O)_3$	$B(W/O)_4$	$\sigma(W/O)_1$	$\sigma(W/O)_2$	$\sigma(W/O)_3$	$\sigma(W/O)_4$	3	6
	A4	I3	I3	I4	I4	4F4.2				4F6.3				4F4.2					

CAMS Error Data Card 4

	5	9	12	15	19	23	27	31	35	39	45	51	57	63	79	80
						Std. Dev. Proportion Est.				Bias in Proportion Estimate						
-nt	Country	Reg.	Zone	Strata	Sub-strata	$\sigma(P/W)_1$	$\sigma(P/W)_2$	$\sigma(P/W)_2$	$\sigma(P/W)_3$	$B(P/W)_1$	$B(P/W)_2$	$B(P/W)_3$	$B(P/W)_4$		4	6
	A4	I3	I3	I4	I4	4F4.2				4F6.3						

SEE Data Cards (cont'd)

Signature Extension Data Card 1

5	9	12	15	21	27	31	35	39	43	47	51	55	59	63	67	71	75	79	80
					Mult. Std. Dev. Error W/W						Additive Std. Dev. Error W/W								
Country	Reg.	Zone	B_{1W}	B_{2W}	σ_{11W}	σ_{12W}	σ_{13W}	σ_{14W}	σ_{15W}	σ_{16W}	σ_{21W}	σ_{22W}	σ_{23W}	σ_{24W}	σ_{25W}	σ_{26W}		1	7
A4	I3	I3	F6.3	F6.3	6F4.2						6F4.2								

Signature Extension Data Card

5	9	12	15	21	27	31	35	39	43	47	51	55	59	63	67	71	75	79	80
					Mult. Std. Dev. Error W/M						Additive Std. Dev. Error W/M								
Country	Reg.	Zone	B_{1M}	B_{2M}	σ_{11M}	σ_{12M}	σ_{13M}	σ_{14M}	σ_{15M}	σ_{16M}	σ_{21M}	σ_{22M}	σ_{23M}	σ_{24M}	σ_{25M}	σ_{26M}		2	7
A4	I3	I3	F6.3	F6.3	6F4.2						6F4.2								

Signature Extension Data Card 3

5	9	12	15	21	27	31	35	39	43	47	51	55	59	63	67	71	75	79	80
					Mult. Std. Dev. Error W/O						Additive Std. Dev. Error W/O								
Country	Reg.	Zone	B_{1O}	B_{2O}	σ_{11O}	σ_{12O}	σ_{13O}	σ_{14O}	σ_{15O}	σ_{16O}	σ_{21O}	σ_{22O}	σ_{23O}	σ_{24O}	σ_{25O}	σ_{26O}		3	7
A4	I3	I3	F6.3	F6.3	6F4.2						6F4.2								

Signature Extension Data Card 4

5	9	12	15	21	27	31	35	39	43	47	51	55	59	63	67	71	75	79	80
					Multiplicative Std. Dev. Error						Additive Std. Dev. Error								
Country	Reg.	Zone	B_1	B_2	σ_{11}	σ_{12}	σ_{13}	σ_{14}	σ_{15}	σ_{16}	σ_{21}	σ_{22}	σ_{23}	σ_{24}	σ_{25}	σ_{26}			
A4	I3	I3	F6.3	F6.3	6F4.2						6F4.2								

SEE Data Cards (cont'd)

2.1.4 Rules for Entering Data Items on Cards

2.1.4.1 General

1. Integers must be right justified.
2. Alphanumeric data must be left justified.
3. F format numbers must have the decimal point present, i.e., F6.2 - NNN.NN. However, the user may override the specified field format as long as the total field width is not exceeded.
4. Except for IYESR, ISIGEX and ICAMER, all other nominal values are zero.

2.1.4.2 Rules for Inputting Specific Fields

- ICSESH must match the case no. of the substrata historical file.
- If IDFRS and IDTOS are left blank, then it will be assumed that the occurrence of any fatal error will terminate the writing of that output file with which it is associated.
- If the first entry of IERDTE data is 0 or blank on card 1 of a YES data group, then the card 2 data is ignored (the card 2 must be present but can be blank except for ID). The IERDTE data and card 2 data from the last non-zero IERDTE data group is used to obtain values for the current group.
- All standard deviation, bias and frequency data items are checked to be within valid ranges. See Section 5.2 for specific field checks.

2.2 FILE INPUT

The only input binary data file is the substrata historical file. See Section 2.4 of the Users Manual for format and contents.

3.0 PROCESSING

3.1 OVERVIEW

The program is basically an input data card error checker and data file generator.

The program will be organized into five basic routines consisting of driver, YES input data processor, CAMS input data processor, signature extension input data processor and error processor. The driver controls the reading of the substrata historical file and the calling of the input processors. Each of the input processors reads an input card group, checks each data field for errors, prints on option the input card image, correlates the substrata historical file with the input data and writes an output file. The error processor catalogs and prints out the input data error messages. The program will continue to read each data type and error check the data until an EOF is reached for that data, regardless of how many errors are detected. If a fatal occurs for a data type, the writing of its output file is terminated. The only exception to this process is that if any input data set is out of order, the program terminates processing and returns control to the system.

3.2 PROCEDURES AND EQUATIONS

There are no application oriented procedures performed except for error checking which is described in Section 5.0.

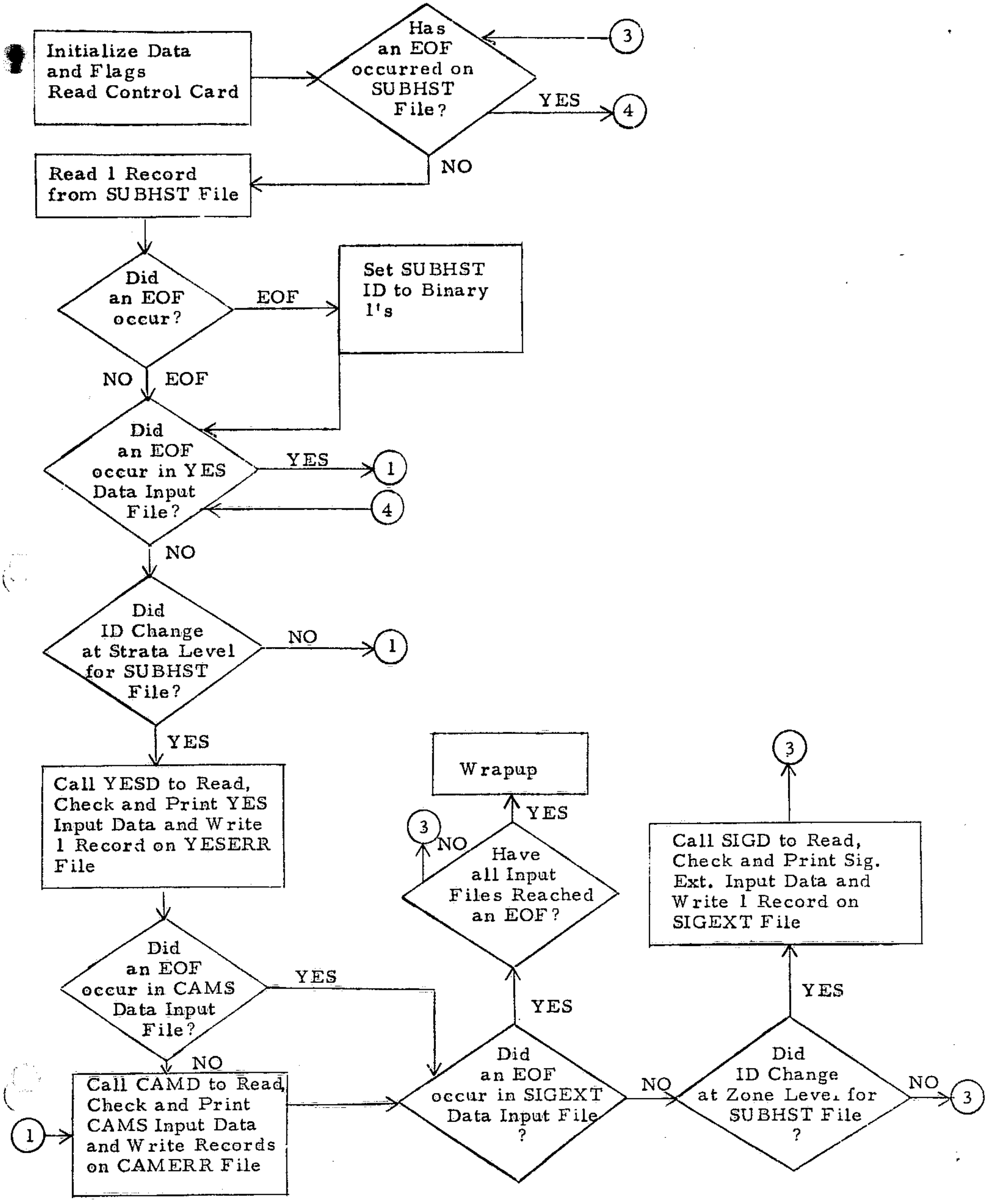
The only calculation performed is that if the country is either USA or CAND, then the yield data is input in bushels per acre and must be converted to quintals per hectar before being written to the YESERR file. The conversion equations are:

1. YIELD = YIELDI / 1.4869664
2. BIASY(I) = BIASYI(I) / 1.4869664 for $1 \leq I \leq 6$
3. SIGYIE(I) = SIGYII(I) / 1.4869664 for $1 \leq I \leq 6$

In addition, the truncation date IERDTE is converted to Zulu date prior to being written on the YESERR file.

3.2 PROGRAM FLOW

Block Diagram



4.0 OUTPUT

There are four media of output:

1. PRINT file
2. YESERR file
3. CAMERR file
4. SIGEXT file

4.1 PRINT DATA

4.1.1 ECHO Print

If ILIST = 1 or 2, then each input data card image will be listed in a form as close to the input form as possible.

Regardless of ILIST's value, if an error is found on a card then the card image is printed in an identical manner.

Since each input data file is processed alternately, the listing of each card type will be intermixed with each other.

Print Format for the Control Card

		IDFRS		IDTOS					
ICASE	ILIST	REG	ZONE	REG	ZONE	IYESR	ISIGER	ICAMR	ICSESH
NNNN	N	NNN	NNN	NNN	NNN	NN	NN	NN	NNNN

Print Format for Bulk Card Data

YES Data Group

← CARD IMAGES FOR 1 ID GROUP → 15
← CARD IMAGES FOR 1 ID GROUP → 25

CAMS Data Group

← CARD IMAGES → 16
↓ 26
36
46

Sig Ext Data Group

← CARD IMAGES → 17
↓ 27
37
47

Repeat for each type and group unless a particular type is omitted via the file unit flag.

4.1.2 Intermediate Debug Print

If ILIST = 2, the contents of each record written on an output file is printed. Again the printout for each file is intermixed in the order of processing. Assuming all three files are being written, the format is as follows:

YESERR File

ICTRY = AAAA IREG = NNN IZONE = NNN ISTRAT = NNNN
Y = NN.NN (IERDTE(I), BIASY(I), SIGYIE(I), I = 1, 6) =
NNNNN NNN.N NN.NN NNNNN ⟶ etc.

CAMERR File

ICTRY = AAAA IREG = NNN IZONE = NNN ISTRAT = NNNN ISUB = NNNN ISEG = NNNN
(PWW(I), PWM(I), PWO(I), BWW(I), SIGWW(I), BWO(I), SIGWO(I),
BWM(I), SIGWM(I), BPW(I), SIGPW(I), I = 1, 4) =
N.NN N.NN N.NN NN.NNN N.NN NN.NNN N.NN NN.NNN N.NN NN.NNN N.NN

Second set is on same line.
Third and 4th set is on 5th line.

SIGEXT File

ICTRY=AAAA IREG = NNN IZONE = NNN
B1W = NN.NNN B2W = NN.NNN SIG1W = $N.NN_1$ --- $N.NN_6$ SIG2W = $N.NN_1$, -- $N.NN_6$
B1M = NN.NNN B2M = NN.NNN SIG1M = $N.NN_1$ --- $N.NN_6$ SIC2M = $N.NN_1$, -- $N.NN_6$
B1O = NN.NNN B2O = NN.NNN SIG1O = $N.NN_1$ --- $N.NN_6$ SIG2O = $N.NN_1$, -- $N.NN_6$
B1 = NN.NNN B2 = NN.NNN SIG1 = $N.NN_1$ --- $N.NN_6$ SIG2 = $N.NN_1$ --- $N.NN_6$

4.1.3 Status Information

The following information will print out at the end of the run.

NNNN SUBSTRATA HISTORICAL FILE RECORDS READ
NNNN YES DATA CARDS READ
NNNN CAMS ERROR DATA CARDS READ
NNNN SIGNATURE EXTENSION DATA CARDS READ
NNNN RECORDS WRITTEN ON YESERR FILE. ICSEYM = NNNN
NNNN RECORDS WRITTEN ON CAMERR FILE. ICSECE = NNNN
NNNN RECORDS WRITTEN ON SIGEXT FILE. ICSESE = NNNN
NNN INPUT ERRORS DETECTED

Either one or the other message will print

JOB COMPLETED

FATAL ERROR OCCURRED IN SEE, FOLLOWING FILES MAY NOT BE ANY GOOD

AAAAAA AAAAAA AAAAAA

4.2 FILE OUTPUT

Up to three files are output from this program.

1. YESERR - YES Error File
2. CAMERR - CAMS Error File
3. SIGEXT - Signature Extension File

See Section 2.4 of the Users Manual for contents and format.

5.0 ERROR PROCESSING

With the exception of the wrong substrata historical file being mounted or any data out of sequence, the program will continue to check for all types of errors even though the writing of a given output file has stopped. There are three categories of errors:

1. Fatal with immediate termination - level 3.
2. Fatal for a given input data type - error checking continues but the writing of the corresponding output file stops - level 2.
3. Nonfatal - normal processing continues - level 1. The card image of the card with field errors or card ID which triggered a general error will be printed prior to the message. The error checks, messages and fatal status are given in Sections 5.1 and 5.2 below.

5.1 GENERAL

Level 3

Code No.

1. A check is made that the correct substrata historical file has been mounted as specified by ICSESH.

   Message:

   EITHER THE SUBSTRATA HISTORICAL FILE HAS NOT BEEN MOUNTED OR IT HAS THE WRONG CASE NC. PROCESSING IS TERMINATED.

2. A check on sequence is made from country down to the card sequence no. for each active card input set. In addition, a missing sequence no. in a card group with the same ID will cause the same action to be taken.

   Message:

   THE ------ INPUT DATA SET IS OUT OF ORDER. PROCESSING IS TERMINATED.

3. A check is made to make sure the ID in IDTOS is ≥ the ID in IDFRS.

   Message:

   ID IN IDTOS IS NOT GREATER OR = TO IDFRS

The following errors are level 2 if the error occurs when the current ID is in the range specified by IDFRS and IDTOS. Otherwise, it is a level 1 error.

10. A check is made to make sure that each SUBHST record has a corresponding record for each of the active input card sets for YES and Sig. Ext. This situation will only be considered an error in case of Sig. Ext. input if there is at least one segment in the SUBHST record.

    Message:

    Optional line
    LEVEL 2 - WRITING OF ------ FILE TERMINATED
    ICTRY ---- IREG ---- IZONE ---- ISTRAT ---- ISUB ----
    ------ INPUT DATA SET DOES NOT HAVE A RECORD WHICH CORRESPONDS TO SUBHST ID.

11. A check is made to make sure that for each record in SUBHST with one or more sample segments there is a CAMS input card group for each segment.

    Message:

    ICTRY ---- IREG --- IZONE --- ISTRAT ---- ISUB ----
    CAMS INPUT DATA SET DOES NOT EXIST FOR SEGMENT ----
    IN SUBHST RECORD.

Level 1 Error
Code No.

15. A check is made to see if an extra input card group has been input with no match on the SUBHST file.

    Message:

    THE ------ INPUT GROUP DOES NOT MATCH WITH SUBHST FILE AND IS EXTRA DATA

5.2 CARD FIELD ERRORS

The occurrence of any of these errors is level 2 if the error occurs when the current ID is in the range specified by IDFRS and IDTOS. Otherwise, it is considered to be a level 1 error.

Code No.

20. A check is made to make sure the error truncation dates in IERDTE on YES card 1 are in ascending order (if not all 0).

    Message:

    IERDTE ARRAY HAS DATES NOT IN ASCENDING ORDER

21. A check is made to make sure YIELDI on YES card 1 is between 0 and 99.99.

    Message:

    YIELDI IS NOT BETWEEN 0 AND 99.99

22. A check is made to make sure that BIASYI on YES card 2 is between -99.9 and +99.9.

    Message:

    ABSOLUTE VALUE OF BIASYI IS NOT LESS THAN 99.9

23. A check is made to make sure that SIGYII on YES card 2 is between 0 and +99.9.

    Message:

    SIGYII IS NOT BETWEEN 0 AND 99.9

24. A check is made to make sure that the P(W/i) data for a CAMS card group is between 0 and 1.

    Message:

    PW- IS NOT BETWEEN 0 AND 1

25. A check is made to make certain that B(W/i) and B_{PW} for a CAMS card group is between -9.999 and +9.999.

    Message:

    ABSOLUTE VALUE OF BW- OR BPW IS NOT LESS THAN 9.999

Code No.

26. A check is made to make certain that σ_{PW} for a CAMS card group is between 0 and 9.99.

Message:

SIGW- OR SIGPW IS NOT BETWEEN 0 AND 9.99

27. A check is made to make sure that B_{1i} and B_{2i} data on the signature extension card group is between -9.999 and +9.999.

Message:

ABSOLUTE VALUE OF B1- OR B2- IS NOT LESS THAN 9.999

28. A check is made to be sure that σ_{1i} or σ_{2i} data on the signature extension card group is between 0 and 9.99.

Message:

SIG1- OR SIG2- IS NOT BETWEEN 0 AND 9.99

29. A check is made to make sure the dates in IERDTE are in range. Year must be greater than 64, month must be between 1 and 12 and day must be between 1 and 31.

Message:

IERDTE ARRAY HAS A BAD YEAR, MONTH OR DAY NUMBER

PART II

COMMON BLOCK DEFINITIONS

COMMON STORAGE ALLOCATION

Name FLAG Size ______ Page 1 of 4

Function ______

Name	Dimension	Format	Description	Symbol	Units
ICTRSV					
IREGSV			Saved SUBHST ID		
IZONSV					
IOUT			Unit 6		
INP			Unit 5		
ICASE	1	I	See Problem Description Case No.		
ILIST	1	I	0 - List cards in error only 1 - Echo input card image 2 - Echo input and print output records		
IDFRS	2	I	Starting region + zone for level 2 errors to occur		
IDTOS	2	I	Ending region + zone for level 2 errors to occur		
IYESU	1	I	Input YES unit, nominal value = 1 (- value indicates do not read)		
ISIGEX	1	I	Input Sig. Ext. unit, nominal value = 2 (- value indicates do not read)		
ICAMER	1	I	Input CAMS unit, nominal value = 4 (- value indicates do not read)		
ICSESH	1	I	Case no. of substrata historical file		
ICTRY	1	I	SUBHST ID for current record Country		
IREG	1	I	Region		
IZONE	1	I	Zone		
ISTRAT	1	I	Strata		

COMMON STORAGE ALLOCATION

Name FLAG Size Page 2 of 4

Function

Name	Dimension	Format	Description	Symbol	Units
ISUB	1	I	SUBHST ID for current record Substrata		
IREGP	1	I	Previous SUBHST record ID Region		
IZONEP	1	I	Zone		
ISTRAP	1	I	Strata		
IFATER	1	I	≠0 indicates a level 3 error has occurred and job is abandoned		
IZSGC	1	I	Count of no. of segments in a zone		
ITYPE	1	I	1 - YES error, 2 - CAMS error, 3 - Sig. Ext. error, error type		
IFATEY	1	I	≠0 indicates a level 2 error has occurred for a particular input - YES input data		
IFATEC	1	I	CAMS error input data		
IFATES	1	I	Signature extension input data		
IRANGE	1	I	=0, no level 2 error can occur ≠0, SUBHST ID in range for level 2		
ICAMR	1	I	=0, read CAM input file		
IYESR	1	I	≠0, don't read YES input file		
ISIGR	1	I	Sig. Ext. input file		
IERR	1	I	Count on no. of errors which have occurred		
IPRINT	1	I	=0, print card image for this error ≠0, card image already printed		
IEFSB	1	I	=0, end of file has not occurred on SUBHST ≠0, EOF has occurred		
ISBHS	1	I	Substrata history file unit no. ≡ 3		

COMMON STORAGE ALLOCATION

Name FLAG Size Page 3 of 4

Function

Name	Dimension	Format	Description	Symbol	Units
IERCDE	1	I	Error code no. as defined in Section 5.2, Problem Description		
ISUBCT	1	I	No. of records read from SUBHST file		
IYSCT	1	I	No. of YES data group records read in		
ICMCT	1	I	No. of CAMS data group records read in		
ISGCT	1	I	No. of Sig. Ext. group records read in		
IYSERC	1	I	No. of records written on YESERR file		
ICMERC	1	I	No. of records written on CAMS error file		
ISIGEC	1	I	No. of records written on Sig. Ext. file		
ILEVES	1	I	Save location for substrata file change level indicator - ILEVEL. Used by SIGD.		
IPSEG	1	I	Seg. ID on SUBHST file with no CAMS input rec. for use by ERRPRO		
IYESER	1	I	YES error output unit ≡ 11		
ICAME	1	I	CAMS error output unit ≡ 8		
ISIGE	1	I	Sig extension output unit ≡ 9		
IREG2P	1	I			
IZON2P	1	I			
ISTR2P	1	I	Previous ID for CAMS input file		
ISUB2P	1	I			
ISEG2P	1	I			

COMMON STORAGE ALLOCATION

Name FLAG Size ______ Page 4 of 4

Function ______

Name	Dimension	Format	Description	Symbol	Units
IREG1P	1	I	Previous ID for YES input file		
IZON1P	1	I			
ISTRIP	1	I			
IREG3P	1	I	Previous ID for Sig. Ext. input file		
IZON3P	1	I			
ILEVEL	1	I	=1 - highest ID level chg substr. =2 - highest ID level chg strata =3 - highest ID level chg zone =4 - highest ID level chg region for SUBHST file		
IFILT1	1	I	For CAMS error data input =0 - constant for segment or none =1 - constant for substrata on down =2 - constant for strata on down =3 - constant for zone on down =4 - constant for region on down		
IFILT2	1	I	For Sig. Ext. same meaning except 0 - means none; 3 - zone; 4 - region		

COMMON STORAGE ALLOCATION

Name SIGIN Size ______ Page 1 of 2

Function Contains data for one signature extension input/output group

Name	Dimension	Format	Description	Symbol	Units
ICTRY3	1	I	Country		
IREG3	1	I	Region		
IZONE3	1	I	Zone		
B1W	1		See Problem Description		
B2W	1				
SIG1W	6				
SIG2W	6				
B1M	1				
B2M	1				
SIG1M	6				
SIG2M	6				
B1O	1				
B2O	1				
SIG1O	6				
SIG2O	6				
B1	1				
B2	1				
SIG1	6				

COMMON STORAGE ALLOCATION

Name SIGIN Size ______ Page 2 of 2

Function ______

Name	Dimension	Format	Description	Symbol	Units
SIG2	6		See Problem Description		

COMMON STORAGE ALLOCATION

Name CAMSIN Size ______ Page 1 of 1

Function Contains data for one CAMS input/output group

Name	Dimension	Format	Description	Symbol	Units
ICTRY2	1	I	See Problem Description		
IREG2	1	I			
IZONE2	1	I			
ISTRA2	1	I			
ISUB2	1	I			
ISEG2	1	I	Segment ID		
PWN	4	Flt.			
PWM	4				
PWO	4				
BWW	4				
BWM	4				
BWO	4				
SIGWW	4				
SIGWM	4				
SIGWO	4				
BPW	4				
SIGPW	4				

COMMON STORAGE ALLOCATION

Name YESIN Size ______

Function Contains data for one YES input/output group with = ID

Name	Dimension	Format	Description	Symbol	Units
ICTRY1			Country		
IREG1			Region		
IZONE1			Zone		
ISTRA1			Strata		
YIELDI	1	Flt.	YIELD - See Problem Description		
IERDTE	3x6	I	Year, month and day for six error truncation dates; see Problem Description		
BIASYI	6	Flt.	YIELD bias error for dates; see Problem Description		
SIGYII	6	Flt.	YIELD std. dev. for dates; see Problem Description		
YIELD	1	Flt.	YIELD for YESERR file	Y	Quintals/Hectare
ITRDTE	6	I	Truncation date in Zulu for YESERR file	-	Days
BIASY	6	Flt.	YIELD bias for YESERR file	B_Y	Quintals/Hectare
SIGYIE	6	Flt.	YIELD std. dev. for YESERR file	σ_Y	Quintals/Hectare

COMMON STORAGE ALLOCATION

Name SUBHIS Size Page 1 of 1

Function Detail record from substrata historical file; also header record data

Name	Dimension	Format	Description	Symbol	Units
NSEG	1	I	No. of sample segments in this substrata		
IDSEG	150	I	List of segment no's. 0 terminates.		
IGRP	1	I			
PWHTH	1	Flt.			
AREA	1	Flt.			
PWTHT	1	Flt.			
NAGR	1	I			
NAL	1	I			
DELPW	1	Flt.			
DELPM	1	Flt.			
CV	(4)	Flt.			
NAME	2	I	8 character name of SUBHST file		
ICSE	1	I	Case no. of SUBHST file		
IMXSEG	1	I	Maximum no. of segment no's in IDSEG		

PART III

LIST OF SUBROUTINES AND SUBROUTINE CALL STRUCTURE

C-4

LIST OF SUBROUTINES IN SEE

Name	Function
SEE	Main control routine for the program. Performs initialization and wrapup functions.
SEECON	Reads the SUBHST file and controls the reading and processing of the three input data files.
YESD	Reads a YES data card group, checks the data for errors and writes a record on the YESERR file.
SIGD	Reads a signature extension ID group and checks for errors and writes the record on the CAMERR file.
CAMD	Controls the reading of CAMS input data groups for all segments with = ID to SUBHST file ID.
SEGPRO	Reads one CAMS error input ID group into CAMSIN COMMON and checks it for errors.
ERRPRO	Controls the processing of all errors, prints error messages and card images according to error type.
LFPA	Given month, day and year, computes Zulu date.

SUBROUTINE CALL STRUCTURE

```
SEE
    ERRPRO
    SEECON
        YESD
            LFPA
            ERRPRO
        CAMD
            ERRPRO
            SEGPRO
                ERRPRO
        SIGD
            ERRPRO
```

PART IV

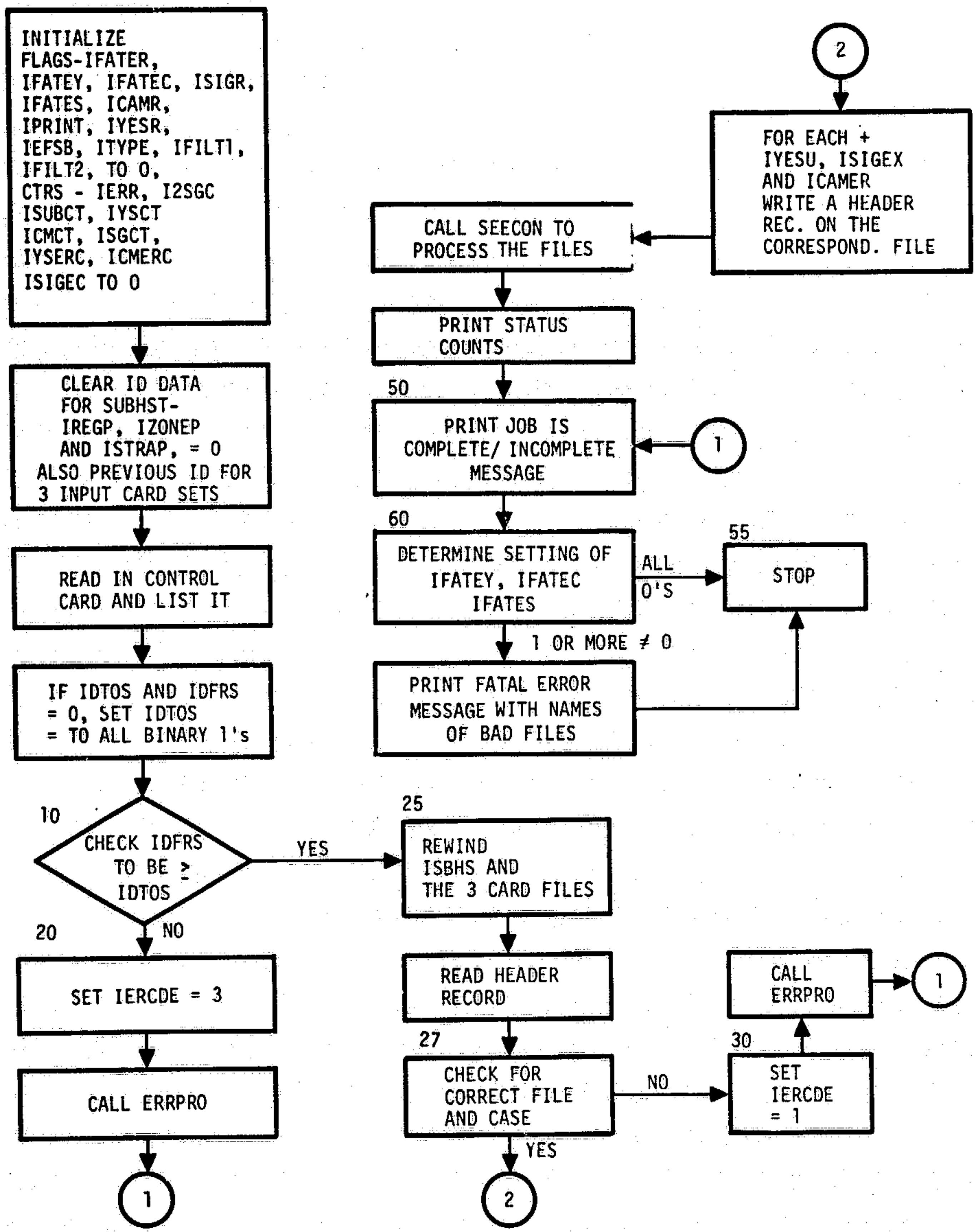
SUBROUTINE DESCRIPTIONS AND FLOWCHARTS

SEE MAIN ROUTINE

Purpose:

Provide main control for the SEE program. It initializes data, reads in control card, lists and checks it, checks for correct substrata historical file to be mounted, calls the input file processing routine SEECON and prints job status information.

MAIN ROUTINE FLOWCHART



SUBROUTINE SEECON

Purpose:

This subroutine reads the substrata historical fil e and controls the reading of the three input data files via subroutine calls. It determines when and if a particular input processor subroutine is to be called.

Input:

FLAG COMMON

IYESU, ISIGEX, ICAMER, ICTRY, IREG, IZONE, ISTRAT, ISUB, IREGP, IZONEP, ISTRAP, IFATEY, IFATEC, IFATES, IEFSB, ISBHS, ISUBCT, IDFRS, IDTOS

SUBHIS COMMON

NSEG, IDSEG, IMXSEG

Output:

FLAG COMMON

ICTRY, IREG, IZ[illegible]E, ISTRAT, ISUB, IZSGC, IRANGE, LZONE, LSTRAT, IPRINT, IEFSB, IERCDE, ISUBCT

Linkage:

CALL SEEPRO

Subroutines Used:

CALL YESD
CALL CAMD
CALL SEGD
CALL ERRPRO

Local Variable Description:

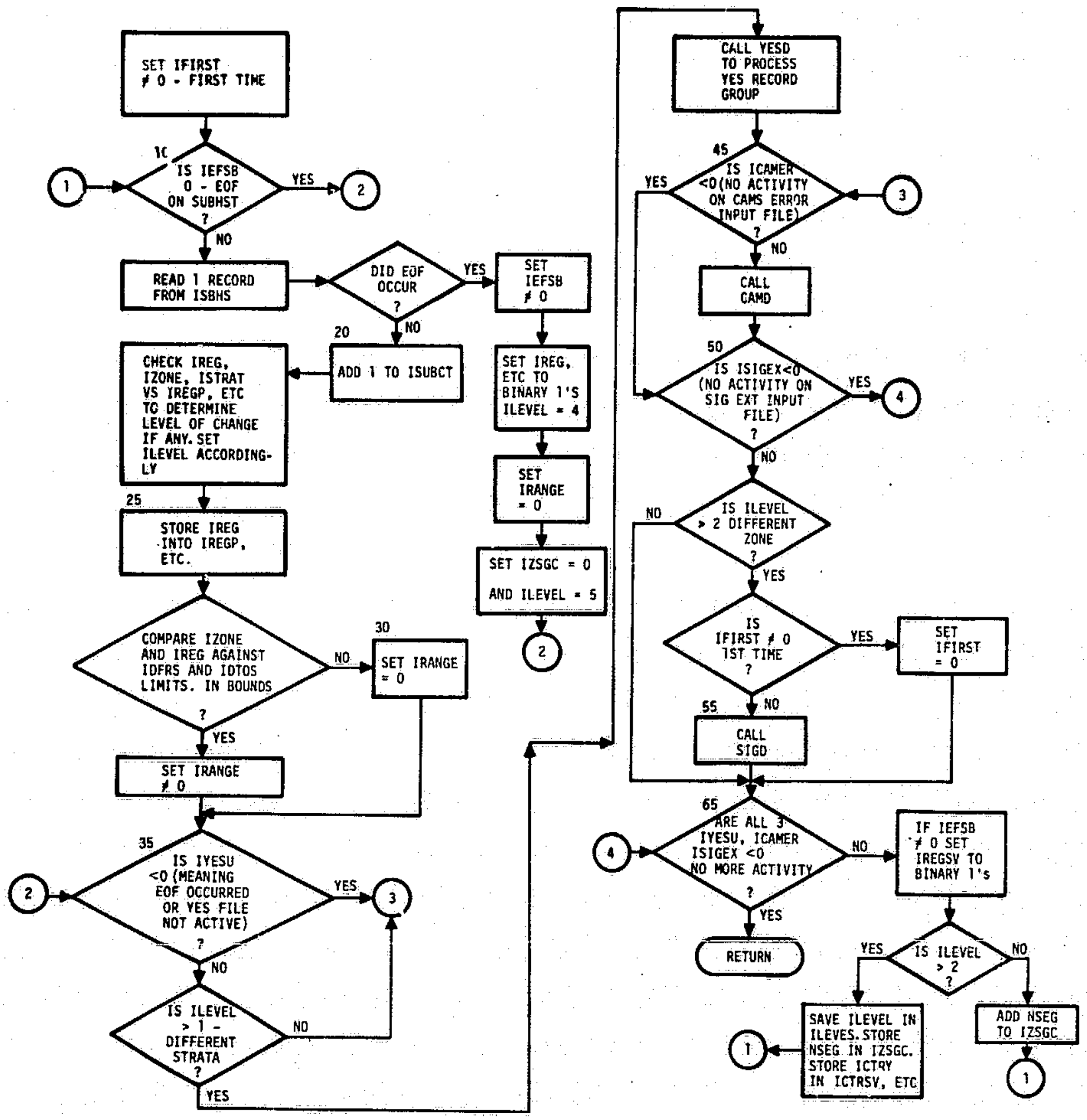
IFIRST - First time flag

$\neq 0$ - first time, $=0$ - not first time

Processing:

See flowchart.

SUBROUTINE SEECON FLOWCHART



ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE YESD

Purpose:

To read a YES record group, list the card images on option, check the data for field errors, check for correct order, determine if current YES ID is compatible to ID of SUBHST, write a record on the YESERR file and indicate when processing for this file is complete. This routine will allow duplicate data groups for any ID level to be entered only once.

Input:

FLAG COMMON
ICTRY, IREG, IZONE, ISTRAT, IFATEY, IYSCT, IYSERC, ILIST, IYESR, IYESER, IYESU, IREG1P, IZON1P, ISTR1P

YESIN COMMON
ICTRY1, ISTRA1, YIELDI, IERDTE, BIASYI, SIGYII

Output:

YESIN COMMON
ID and YIELD, ITRDTE, BIASY, SIGYIE

FLAG COMMON
ITYPE, IPRINT, IERCDE, IYSCT, IYSERC, IYESR, IREG1P, IZON1P, ISTR1P

Linkage:

CALL YESD

Subrouitnes Used:

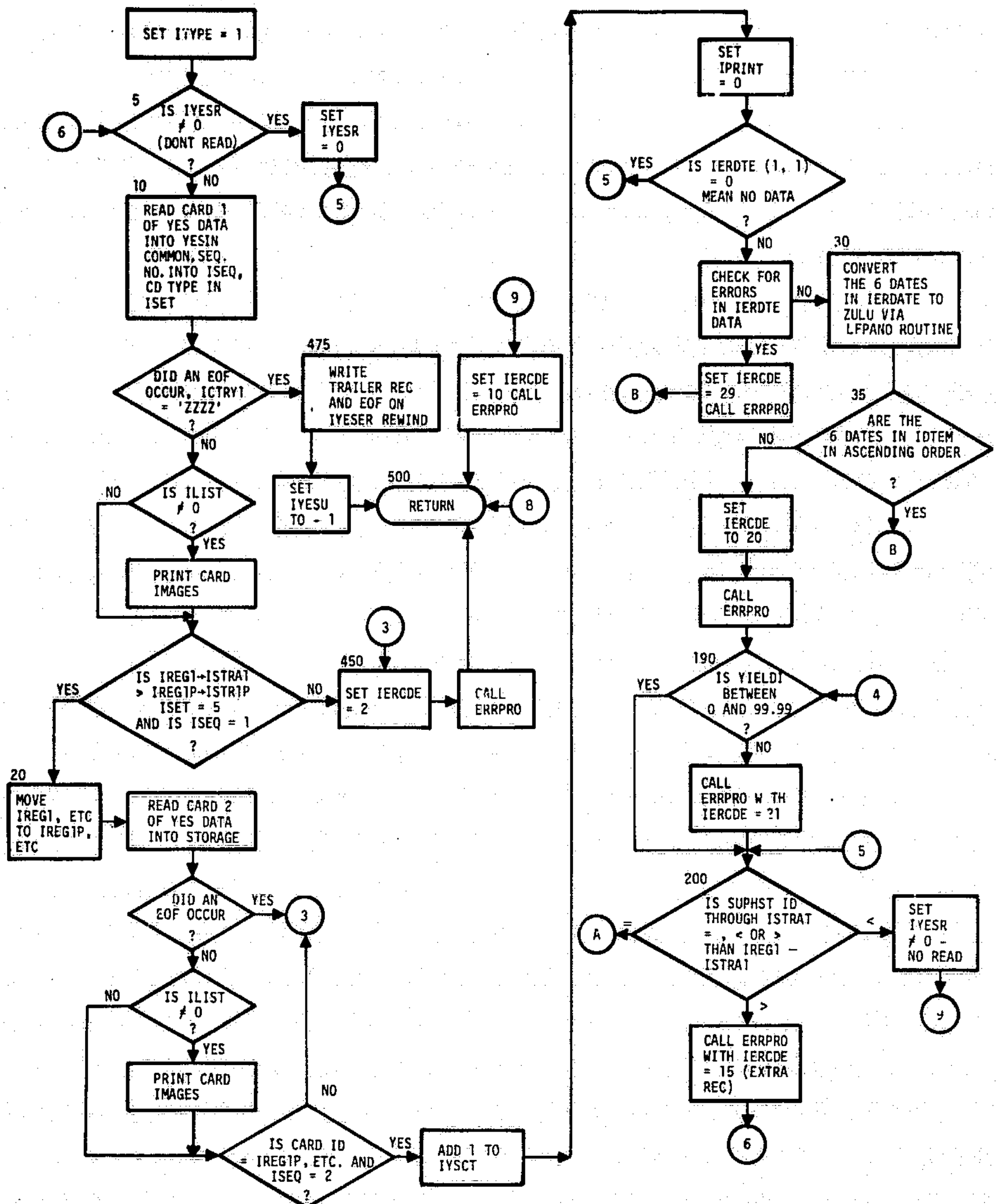
CALL LFPA (day, month, year, dum, days)
Given day, month, year, compute days in Zulu.
CALL ERRPRO

Local Variable Descriptions:

ISEQP	Previous sequence no. for cards
ISEQ	Current card seq. no.
ISET	Card set no. should ≡ 5
IDTEM(6)	Temporary Zulu IERDTE loc.

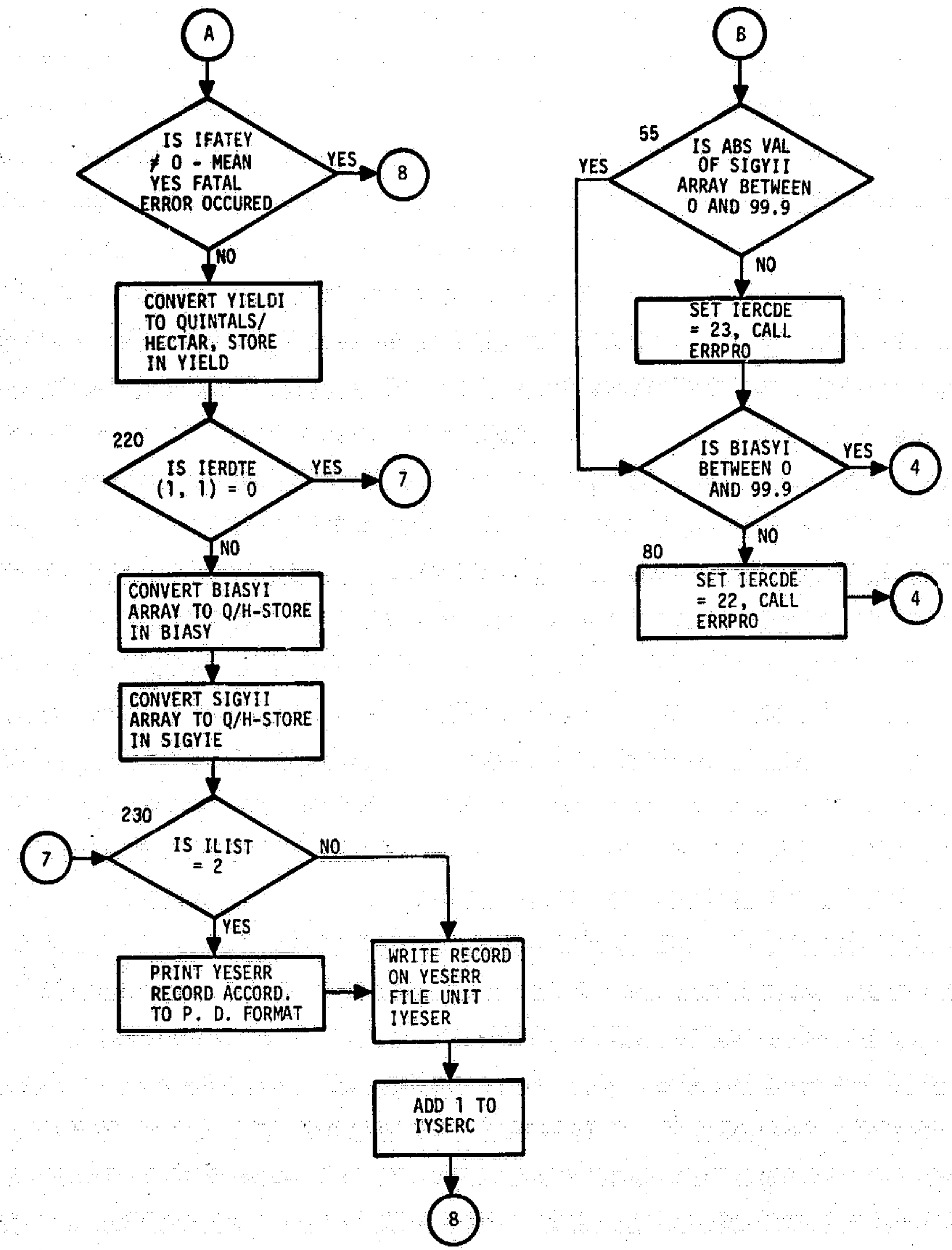
Processing: See flowchart.

YESD FLOWCHART



ORIGINAL PAGE IS OF POOR QUALITY

YESD FLOWCHART (CONTINUED)



SUBROUTINE SIGD

Purpose:

This routine reads a signature extension record group, lists the card images on option, checks the data for field errors and correct order. It determines if SIG. EXT. ID is compatible to SUBHST ID, writes a record on the SIGEXT file and indicates when processing for this file is complete. This routine will allow duplicate data groups for any ID level to be entered once.

Input:

FLAG COMMON

ICTRSV, IREGSV, IZONSV, IFATES, ISGCT, ISIGEC, ILIST, ISIGR, ISIGEX, ISIGE, ILEVEL, IFILT2, IZSGC, IREG3P, IZON3P

SIGIN COMMON

ICTRY3, IREG3, IZONE3, plus all signature extension data.

Output:

FLAG COMMON

ITYPE, IPRINT, ISGCT, ISIGEC, IREG3P, IZON3P, IERCDE, ISIGR

SIGIN COMMON

All data.

Linkage:

CALL CAMD

Subroutines Used:

CALL ERRPRO

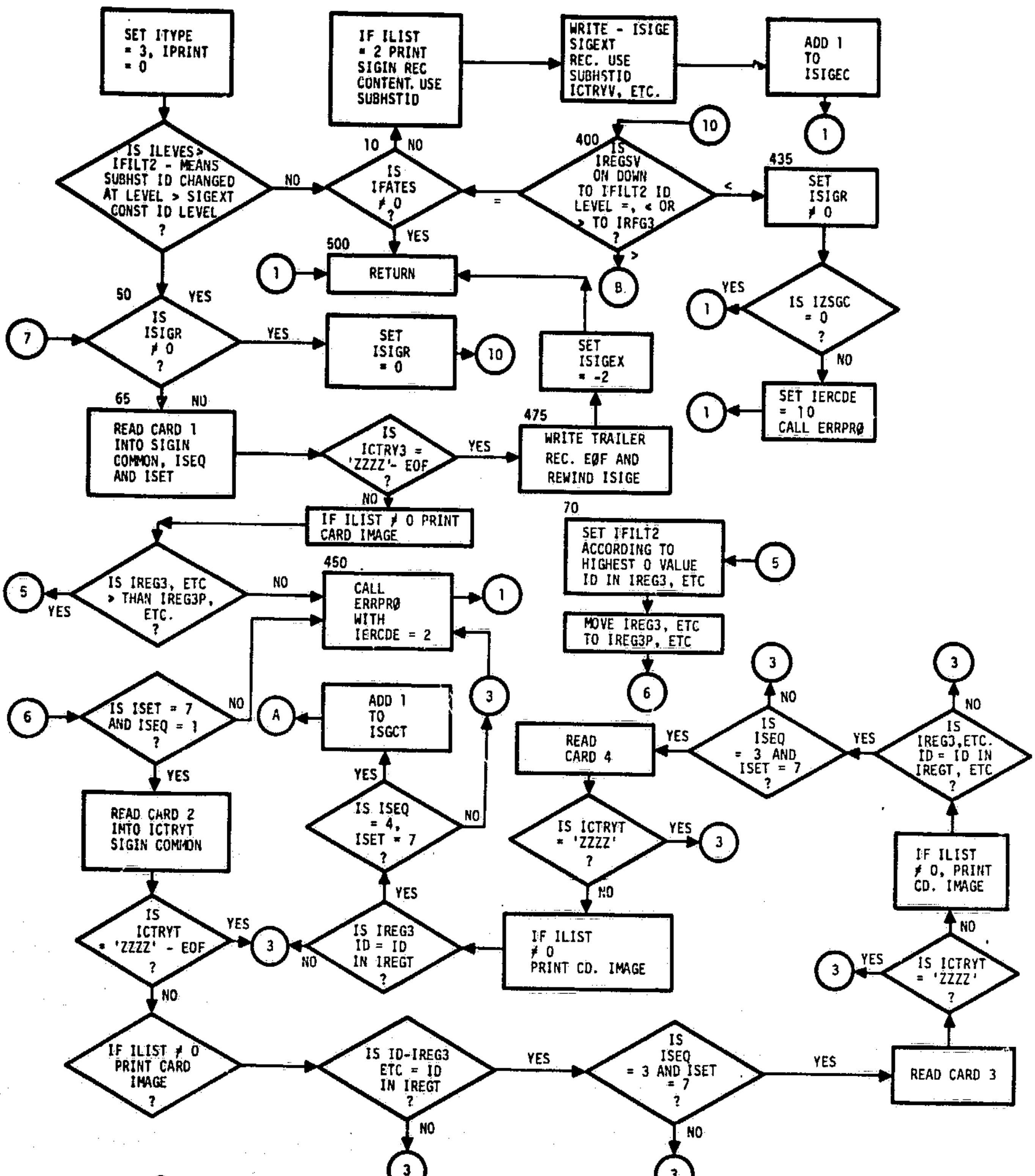
Processing:

See flowchart.

Local Variable Descriptions:

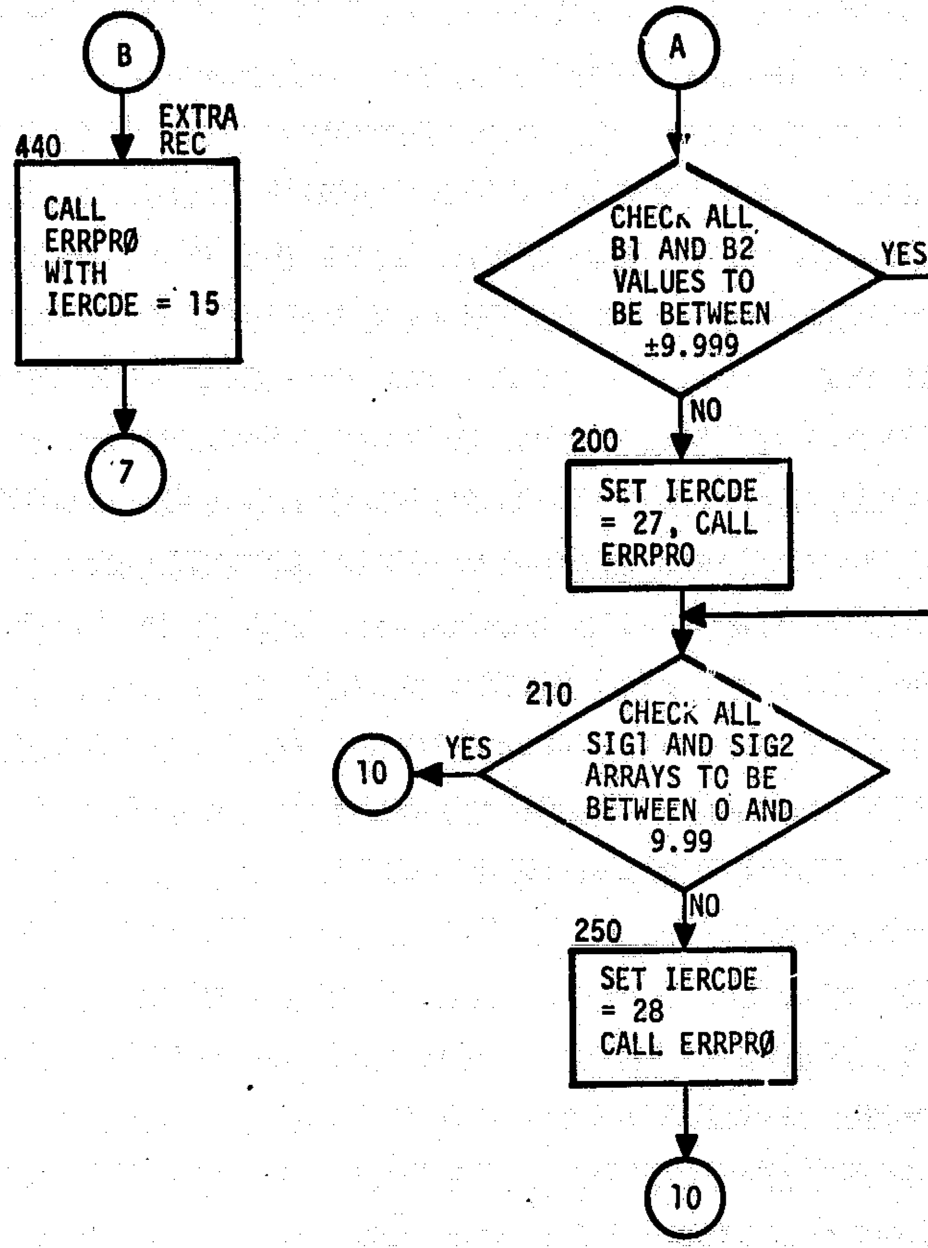
ICTRYT, IREGT, IZONET - temporary ID locations.

SIGD FLOWCHART



ORIGINAL PAGE IS OF POOR QUALITY

SUBROUTINE SIGD FLOWCHART (CONT'D)



SUBROUTINE CAMD

Purpose:

To control the reading of CAMS input data groups for all segments with = ID to SUBHST file. Check for data to be in order and not missing; also check for ID compatibility with SUBHST file (accounting for multilevel constant data). Make sure there is a segment for each segment in SUBHST file substrata record. Write a record on the CAMERR file and indicate when processing for this file is complete.

Input:

FLAG COMMON
ICTRY, IREG, IZONE, ISTRAT, IFATEC, ICAMR, ICMERC, ILEVEL, IFILT1, ICAME, ILIST, ICAMER, IREG2P, IZON2P, ISTR2P, ISUB2P, ISEG2P

CAMSIN COMMON
ICTRY2, IREG2, IZONE2, ISTRA2, ISUB2, ISEG2, and all application data

SUBHIS COMMON
NSEG, IDSEG

Output:

FLAG COMMON
IPSEG, ITYPE, IERCDE, ICMERC, ICAMR

CAMSIN COMMON
All data

Linkage:

CALL CAMD

Subroutines Used:

CALL SEGPRO (IND, IOPT) to read in 1 data group for a segment and do field/seq check.

IND =0 = SUBSTRA ID
$\neq$0 $\neq$ SUBSTRA ID or EOF

IOPT =0 - Full process with ID change
$\neq$0 - Ignore data with ID change

CALL ERRPRO

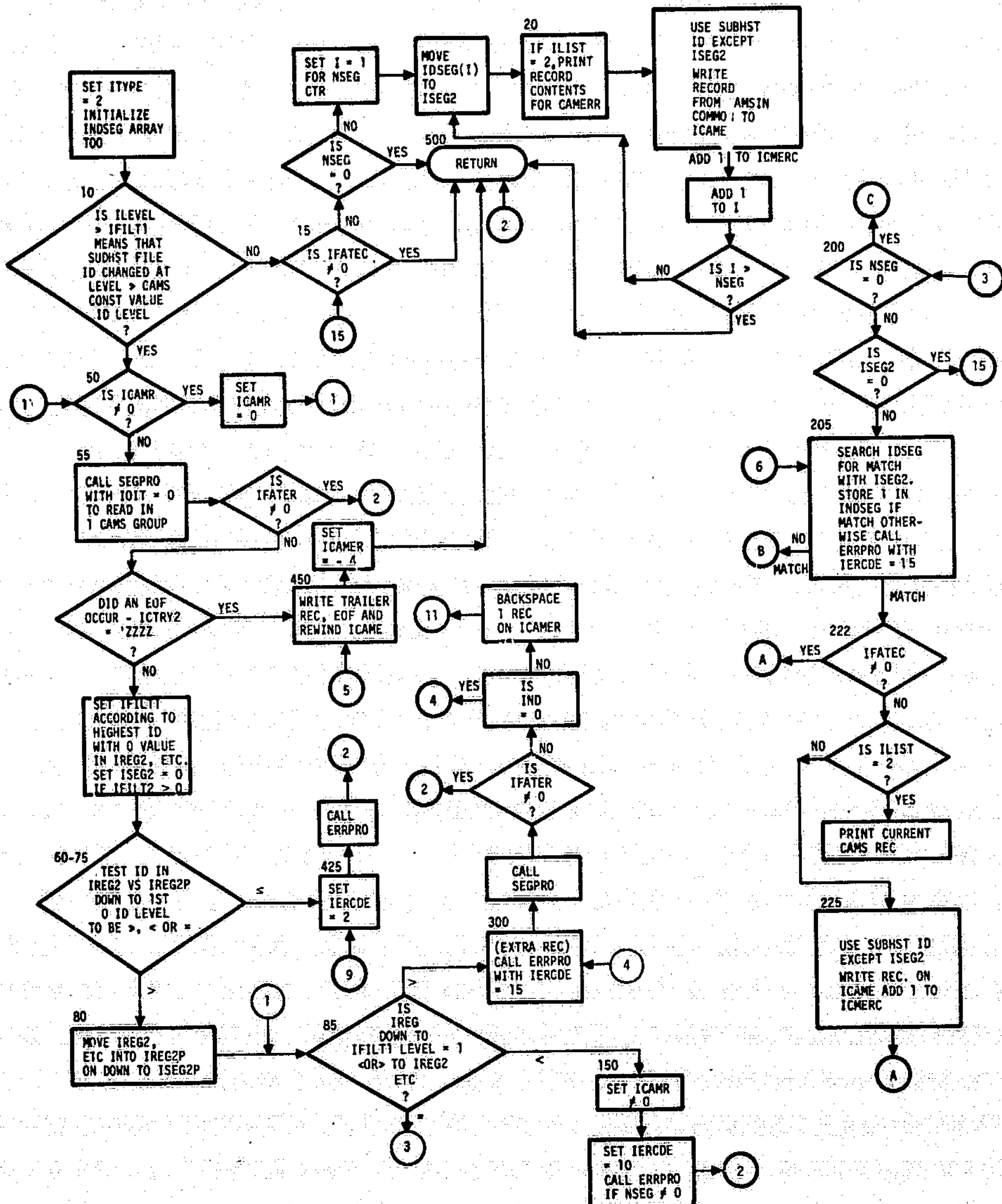
Processing:

See flowchart.

Local Variable Descriptions:

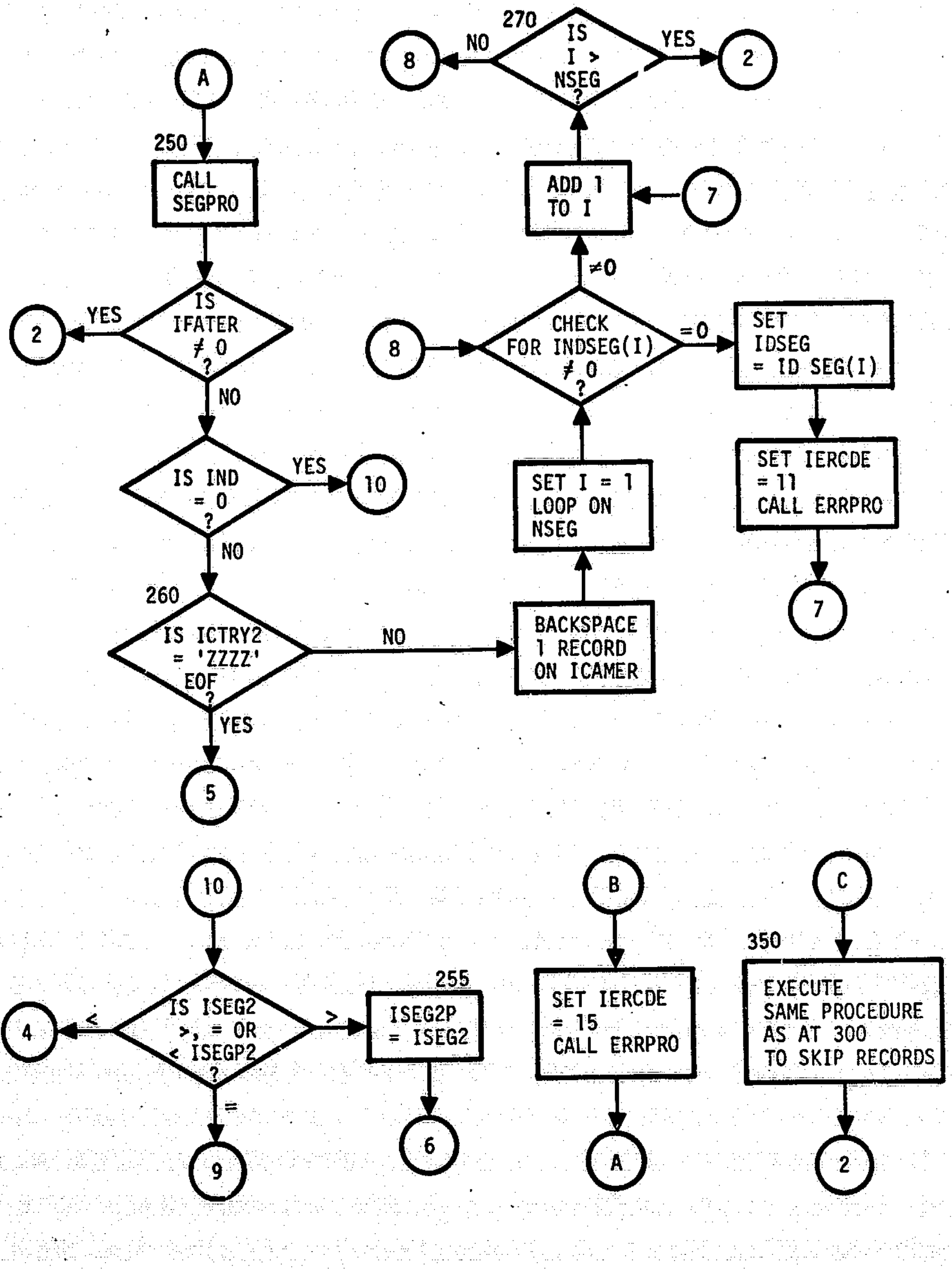
INDSEG (150) - Indicator array 1 to 1 with segment list in SUBHST record - 0 $\rightarrow$ no match on CAM input; $\neq$0 - means seg ID matched.

SUBROUTINE CAMD FLOWCHART



ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE CAMD FLOWCHART (CONTINUED)



SUBROUTINE SEGPRO (IND, IOP)

Purpose:

To read 1 CAMS error input ID group into CAMSIN COMMON, determine if new ID has been read in; if so, set IND ≠ 0. If not, check order and completeness within group and perform field checks. Print card images of group if ILIST ≠ 0.

Input:

FLAG COMMON

ILIST, IREG2P, IZON2P, ISTRA2P, ISUB2P, ICAMER, ICMCT, ISEQ, ISET

CAMSIN COMMON

All data

Output:

CAMSIN COMMON

All data

FLAG COMMON

IPRINT, IERCDE, ICMCT

Linkage:

CALL SEGPRO (IND, IOPT)

On exit	-	IND = 0	No change in ID sort field
		IND ≠ 0	Change in ID sort fields
On entry	-	IOPT = 0	Means an initial substrata call; full process if change in ID
		IOPT ≠ 0	If ID change, set IND ≠ 0 and exit immediately

Subroutines Called:

CALL ERRPRO

Processing:

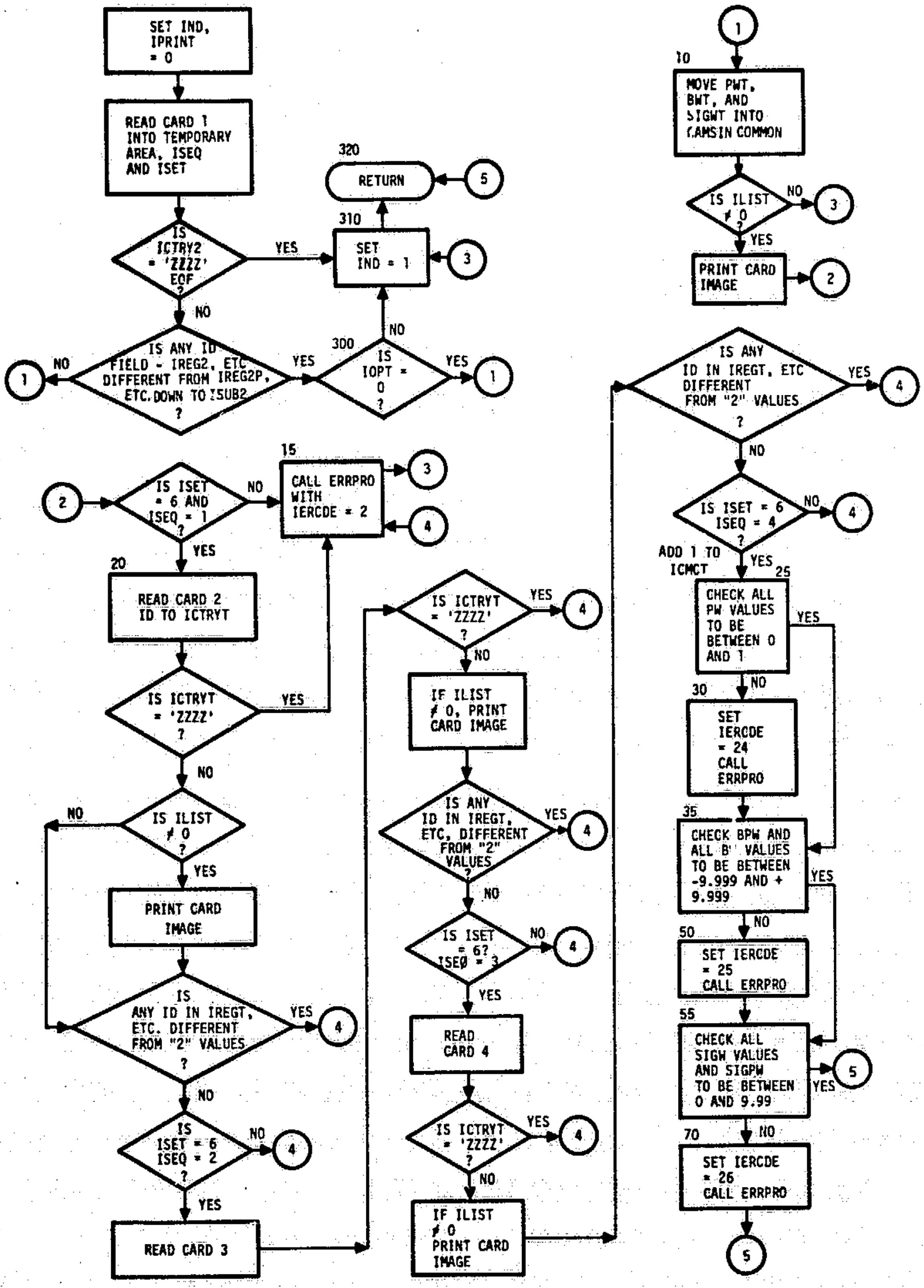
See flowchart.

Local Variable Descriptions:

IREGT, IZONET, ISTRTT, ISUBT, ISEGT, ICTRYT - temporary ID locations

PWT(4), BWT(4) and SIGWT(4) - temporary location for card 1 data

SUBROUTINE SEGPRO FLOWCHART



SUBROUTINE ERRPRO

Purpose:

To process errors by category; setting status, initialing wrapup, printing messages and printing card images according to the category.

Input:

FLAG COMMON

IPSEG, IERR, IERCDE, IPRINT, ITYPE, IRANGE, ILIST, IYESER, ICAME, ISIGE, ICTRSV, IREGSV, IZONSV, ICTRY, IREG, IZONE, ISTRAT, ISUB

All data in CAMSIN and SIGIN COMMONS and all data except YIELD, ITRDTE, BIASY and ISGYIE in YESIN COMMON.

Output:

FLAG COMMON

IFATER, IFATEY, IFATEC, IFATES, IERR, IPRINT, IYESU, ISIGEX, ICAMER

Messages as described in problem description:

DATA PRINT OUT FOR ERROR CODES 2, 10-19 ASSUMING ILIST = 0, OTHERWISE JUST MESSAGE PRINTS.

Prints:

ICTRY ---- IREG --- IZONE --- ISTRAT ---- ISUB ---- ISEG ----

Message:

DATA PRINT OUT FOR ERROR CODES 20-28 ASSUMING ILIST = 0, OTHERWISE JUST MESSAGE PRINTS.

ICTRY IREG IZONE ISTRAT ISUB ISEG card image 1 N
card image 2 N
card image n N

Linkage:

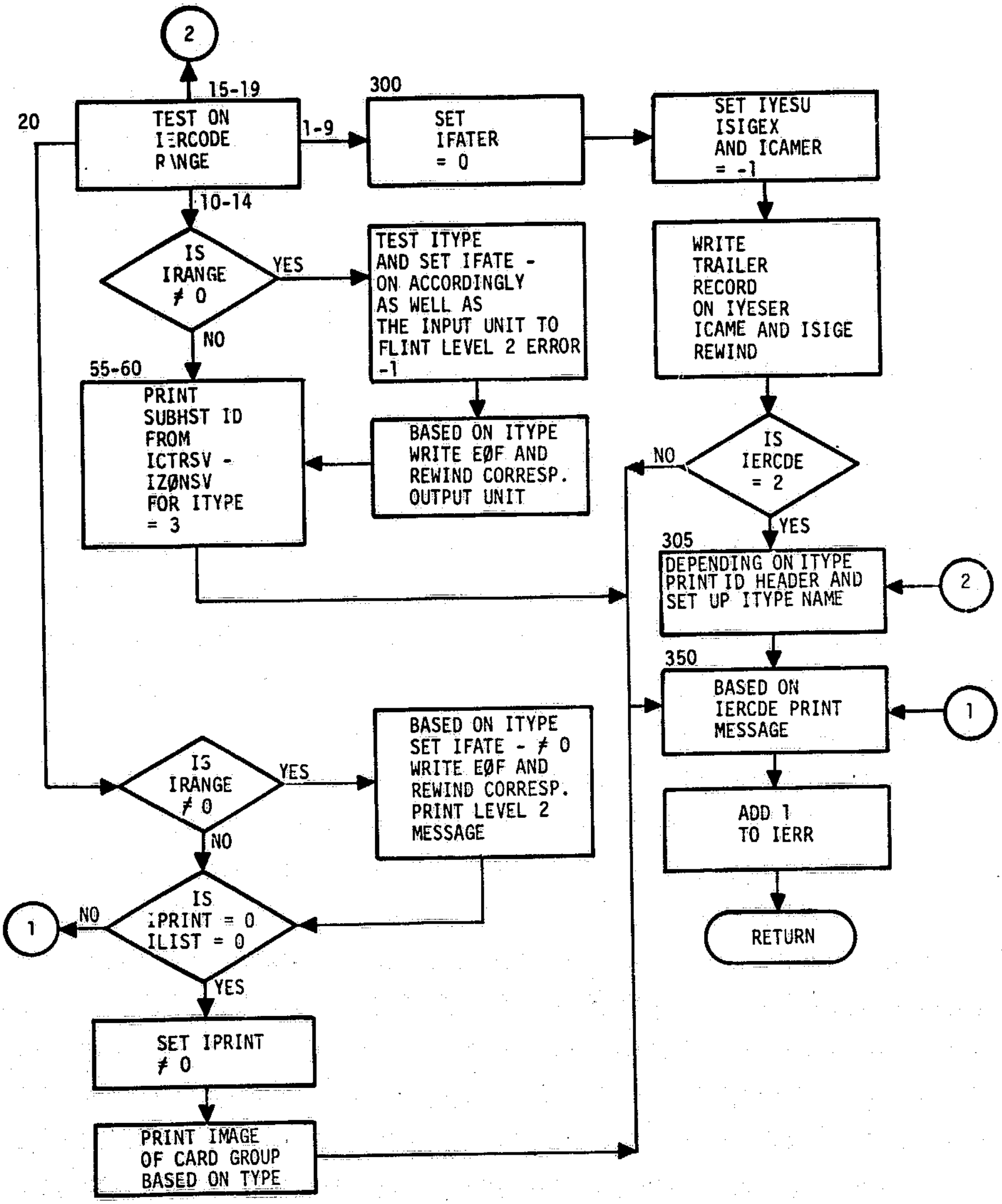
CALL ERRPRO

Subroutines Used:

None.

Processing: See flowchart.

SUBROUTINE ERRPRØ FLOWCHART



SUPPLIED UTILITY ROUTINES

Routine Day

Call Day (IYMD, IDAY)

Given IYMD (3) where	IYMD (1)	IS Day No.
	IYMD (2)	IS Month No.
	IYMD (3)	IS Year No.

Compute year day no. in IDAY

Routine PIMOD

Call PIMOD (A)

Convert ±A in radians to an angle $0-2\pi$

Routine SOL (Entry ALPHA)

Call ALPHA (IFLAG)

For emphemeris usage as called by hector

computes ALPHAM and ALPHAT and IFLAG = 1

Routine PAGER (Entry Eject)

Call PAGER (NLINES)

Updates line count in NLINE with NLINES

NPAGE = 0 causes page to be restored prior to print.

NPAGE - page no.

HEADER- 80 char. 20A5

ICASE- case no.

KO - 6 print unit

INMAX is max no. of lines allowed

Initially NLINE should be set > LINMAX and NPAGE = 0

SUPPLIED UTILITY ROUTINES
(CONTINUED)

Call EJECT (NLINES)

Causes page to be restored automatically and then prints headers.

Routine CLDAY

Call CLDAY

Given IDAY-DAY no. of the year compute in LMO-the month and in LDA the day no.
Need: IYEAR = 0 - Leap Year, ≠ 0 not Leap Year

Routine KEPLER

Call KEPLER (XM, XECC, XE, ERROR)

Given XM - Mean anomaly, XECC - eccentricity
Compute: E-eccentric anomaly, error = 0 means OK

Routine LFPA

Call LFPA [FLDA, LMO, LYR, ALFGM (can be dummy), DAYS]

Given: FLDA - day of month no., LMO - month no.,
LYR - year no. compute ALFGM - right ascension and
DAYS - Zulu day no.

Routine DEGMOD

Call DEGMOND (RAD, IDEG)

Given: angle rad in radians store the angle in deg., min., sec., in IDEG(1) - (3).

Routine FZULU

Call FZULU (IOATE, IOUT)

Given Zulu date in IDATE, compute year, month and day in IOUT(1) - IOUT(3).

Routine RDMIA

Call RDMIA(FL, U)

Given double precision random no. seed in FL, comoute random no. U (0-1) based on uniform distribution.

PART V

SUBROUTINE LISTINGS

```
      BLOCK DATA                                                          BLOCK
      COMMON /FLAG/                                                       FLAG
     1ICTRSV,IREGSV,IZONSV,INP,IOUT,ICASE,IHIST,IDFRS(2),IOTOS(2),        FLAG
     2IYESU,ISIGEX,ICAMER,ICSFSH,ICTRY,IREG,IZONE,ISTRAT,ISUB,IREGP,      FLAG
     3IZONEP,ISTRAP,IFATER,IZSGC,ITYPE,IFATFY,IFATEC,IFATES,IRANGE,       FLAG
     4ICAMR,IYESR,ISIGR,IFRR,IPRINT,IFFSH,ISHHS,IFRCDE,ISUBCT,            FLAG
     5IYSCT,ICMCT,ISGCT,IYSFRC,ICMFRC,ISIGEC,IPSEG,IYFSFR,ICAME,          FLAG
     6ISIGE,IREG2P,IZON2P,ISTR2P,ISUB2P,ISEG2P,IREG1P,IZON1P,ISTR1P,      FLAG
     7IREG3P,IZON3P,ILEVEL,IFILT1,IFILT2                                  FLAG
     8,ILEVFS                                                             MOD3
      COMMON /SUBHIS/                                                     SUBHIS
     1NSEG,IDSEG(150),IGRP,PWHTH,AREA,PWTHT,NAGR,NAL,DELPW,DELPH,         SUBHIS
     2CV(4),NAME(2),ICSF,IMXSEG                                           SUBHIS
      COMMON /YESIN/                                                      YESIN
     1ICTRY1,IREG1,IZONE1,ISTRA1,YIELDI,IERDIE(3,6),BIASYI(6),SIGYII(6),YESIN
     2YIELD,IRDTE(6),BIASY(6),SIGYIE(6)                                   YESIN
      COMMON /CAMSIN/                                                     CAMSIN
     1ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,PWW(4),PWM(4),PWO(4),        CAMSIN
     2BWW(4),BWM(4),BWO(4),SIGWW(4),SIGWM(4),SIGWO(4),BPW(4),SIGPW(4)     CAMSIN
      COMMON /SIGIN/                                                      SIGIN
     1ICTRY3,IREG3,IZONE3,B1W,B2W,SIG1W(6),SIG2W(6),B1M,B2M,SIG1M(6),     SIGIN
     2SIG2M(6),B1O,B2O,SIG1O(6),SIG2O(6),B1,B2,SIG1(6),SIG2(6)            SIGIN
      DATA IYESU,ICAMER,ISIGEX,INP,IOUT,ISHHS,IYESER,ICAME,ISIGE          BLOCK
     1/1,4,2,5,6,3,11,8,9/                                                BLOCK
      END                                                                 BLOCK
```

ORIGINAL PAGE IS
OF POOR QUALITY

@ ELT CAMD,1,760426, 48473 , 1

```
      SUBROUTINE CAMD                                                   CAMD
C     CONTROLS THE READING OF CAMS INPUT DATA GROUPS FOR ALL SEGMENTS   CAMD
C     WITH = ID TO SUBHST FILE ID.                                      CAMD
      COMMON /CAMSIN/                                                   CAMSIN
     1ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,PWW(4),PWM(4),PWO(4),      CAMSIN
     2BWW(4),BWM(4),BWO(4),SIGWW(4),SIGWM(4),SIGWO(4),BPW(4),SIGPW(4)   CAMSIN
      COMMON /FLAG/                                                     FLAG
     1ICTRSV,IREGSV,IZONSV,INP,IOUT,ICASE,ILIST,IDFFS(2),TOTUS(2),      FLAG
     2IYESU,ISIGEX,ICAMER,IPSFSH,ICTRY,IREG,IZONE,ISTRAT,ISUB,IREGP,    FLAG
     3IZONEP,ISTRAP,IFATER,IZSGC,ITYPE,IFATEY,IFATLC,IFATES,IRANGE,     FLAG
     4ICAMR,IYESR,ISIGR,IERR,IPRINT,IFFSH,ISBHS,IERCUE,ISUBCT,          FLAG
     5IYSCT,ICHCT,ISGCT,IYSERC,ICMERC,ISIGEC,IPSEG,IYFSER,ICAME,        FLAG
     6ISIGE,IREG2P,IZON2P,ISTR2P,ISUB2P,ISEG2P,IREG1P,IZON1P,ISTR1P,    FLAG
     7IREG3P,IZON3P,ILEVEL,IFILT1,IFILT2                                FLAG
     8,ILEVEL                                                           MOD3
      COMMON /SUBHIS/                                                   SUBHIS
     1NSEG,IDSEG(150),IGRP,PHHTH,AREA,PHTHT,NAGR,NAL,DELPH,DELPM,       SUBHIS
     2CV(4),NAME(2),ICSF,IMXSEG                                         SUBHIS
      DIMENSION INDSEG(150)                                             CAMD
      DATA IZ/4HZZZZ/,IZAP/0/                                           CAMD
      ITYPL = 2                                                         CAMD
      DO 10 I=1,150                                                     CAMD
      INDSEG(I) = 0                                                     CAMD
   10 CONTINUE                                                          CAMD
      IF(ILEVEL .GT. IFILT1)GO TO 50                                    CAMD
   15 IF(IFATEC .NE. 0)GO TO 500                                        MOD2
      IF(NSEG .EQ. 0)GO TO 500                                          CAMD
      DO 20 K=1,150
      IF(IDSEG(K) .EQ. 0)GO TO 25
      ISEG2 = IDSEG(K)
      IF(ILIST .EQ. 2)WRITE(IOUT,900)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISEG2,CAMD
     1(PWW(I),PWM(I),PWO(I),BWW(I),SIGWW(I),BWM(I),SIGWM(I),BWO(I),     MOD3
     2SIGWO(I),BPW(I),SIGPW(I),I=1,4)                                   CAMD
      WRITE(ICAME)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISEG2,(PWW(I),PWM(I),    CAMD
     1PWO(I),BWW(I),SIGWW(I),BWM(I),SIGWM(I),BWO(I),SIGWO(I),BPW(I),    MOD3
     2SIGPW(I),I=1,4)                                                   CAMD
      ICMERC = ICMERC + 1                                               CAMD
   20 CONTINUE                                                          CAMD
   25 GO TO 500                                                         CAMD
   50 IF(ICAMR .EQ. 0)GO TO 55                                          CAMD
      ICAMR = 0                                                         CAMD
      GO TO 85                                                          CAMD
   55 CALL SEGPRO(IND,0)                                                CAMD
      IF(IFATER .NE. 0)GO TO 500                                        CAMD
      IF(ICTRY2 .EQ. IZ)GO TO 450                                       CAMD
      IFILT1 = 0                                                        CAMD
      IF(ISUB2 .EQ. 0)IFILT1 = 1                                        CAMD
      IF(ISTRA2 .EQ. 0)IFILT1 = 2                                       CAMD
      IF(IZONE2 .EQ. 0)IFILT1 = 3                                       CAMD
      IF(IREG2 .EQ. 0)IFILT1 = 4                                        MOD2
      IF(IFILT1 .NE. 0)ISEG2 = 0                                        CAMD
      IF(IREG2 .LT. IREG2P)GO TO 425                                    CAMD
      IF(IREG2 .EQ. IREG2P)GO TO 60                                     CAMD
      GO TO 80                                                          CAMD
   60 IF(IZONE2 .LT. IZON2P)GO TO 425                                   CAMD
      IF(IZONE2 .EQ. IZON2P)GO TO 65                                    CAMD
      GO TO 80                                                          CAMD
   65 IF(ISTRA2 .LT. ISTR2P)GO TO 425
```

```
      IF(ISTRA2 .EQ. ISTR2P)GO TO 70                                  CAMD
      GO TO 80                                                        CAMD
   70 IF(ISUB2 .LT. ISUB2P)GO TO 425                                  CAMD
      IF(ISUB2 .EQ. ISUB2P)GO TO 75                                   CAMD
   75 IF(ISEG2 .LT. ISEG2P)GO TO 425                                            *NEW
      IF(ISEG2 .EQ. ISEG2P .AND. ISEG2 .NE. 0)GO TO 425                         *NEW
   80 IREG2P = IREG2                                                  CAMD      **-1
      IZON2P = IZONE2                                                 CAMD
      ISTR2P = ISTRA2                                                 CAMD
      ISUB2P = ISUB2                                                  CAMD
      ISEG2P = ISEG2                                                  CAMD
   85 IF(IFILT1 .EQ. 4)GO TO 200                                      CAMD
      IF(IREG - IREG2)150, 90,300                                     CAMD
   90 IF(IFILT1 .EQ. 3)GO TO 200                                      CAMD
      IF(IZONE - IZONE2)150,95,300                                    CAMD
   95 IF(IFILT1 .EQ. 2)GO TO 200                                      CAMD
      IF(ISTRAT - ISTRA2)150,100,300                                  CAMD
  100 IF(IFILT1 .EQ. 1)GO TO 200                                      CAMD
      IF(ISUB - ISUB2)150,200,300                                     CAMD
  150 ICAMR = 1                                                       CAMD
      IERCDE = 10                                                     CAMD
      IF(NSEG .NE. 0)CALL ERRPRO                                      CAMD
      GO TO 500                                                       CAMD
  200 IF(NSEG .EQ. 0)GO TO 350                                        MOD1
      IF(ISEG2 .EQ. 0)GO TO 15                                        MOD2
  205 DO 210 I=1,150                                                  CAMD
      ISAV = I                                                        CAMD
      IF(IDSEG(I) .EQ. 0)GO TO 215                                    CAMD
      IF(IDSEG(I) .EQ. ISEG2)GO TO 220                                CAMD
  210 CONTINUE                                                        CAMD
  215 IERCDE = 15                                                     CAMD
      CALL ERRPRO                                                     MOD2
      GO TO 250                                                       CAMD
  220 INDSEG(ISAV) = 1                                                CAMD
  222 IF(IFATEE .NE. 0)GO TO 250                                      MOD2
      IF(ILIST .NE. 2)GO TO 225                                       CAMD
      WRITE(IOUT,900)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISEG2,(PWW(I),PWM(I),  CAMD
     1PWO(I),RWW(I),SIGWW(I),RWM(I),SIGWM(I),RWO(I),SIGWO(I),RPW(I),  MOD3
     2SIGPW(I),I=1,4)                                                 CAMD
  225 WRITE(ICAME)ICTRY,IREG,IZONE,ISTRAT,ISUB,ISEG2,(PWW(I),PWM(I),  CAMD
     1PWO(I),PWW(I),SIGWW(I),RWM(I),SIGWM(I),RWO(I),SIGWO(I),RPW(I),  MOD3
     2SIGPW(I),I=1,4)                                                 CAMD
      ICMERC = ICMERC + 1                                             CAMD
  250 CALL SEGPRO(IND,I)                                              CAMD
      IF(IFATER .NE. 0)GO TO 500                                      CAMD
      IF(IND .NE. 0)GO TO 260                                         CAMD
      IF(ISEG2 - ISEG2P)425,425,255                                   CAMD
  255 ISEG2P = ISEG2                                                  CAMD
      GO TO 205                                                       CAMD
  260 IF(ICTRY2 .EQ. IZ)GO TO 450                                     CAMD
      BACKSPACE ICAMER                                                CAMD
      IF(ISEG2P .EQ. 0)GO TO 500                                      MOD2
      DO 270 I=1,150                                                  CAMD
      IF(IDSEG(I) .EQ. 0)GO TO 500                                    CAMD
      IF(INDSEG(I) .NE. 0)GO TO 270                                   CAMD
      IPSEG = IDSEG(I)                                                CAMD
      IERCDE = 11                                                     CAMD
      CALL ERRPRO                                                     CAMD
  270 CONTINUE                                                        CAMD
      GO TO 500                                                       CAMD
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
  300 IERCDE = 15                                                       CAMD
      CALL ERRPRO                                                       CAMD
      CALL SEGPRO(IND,1)                                                CAMD
      IF(IEATER .NE. 0)GO TO 500                                        CAMD
      IF(IND .EQ. 0)GO TO 300                                           CAMD
      BACKSPACE ICAMER                                                  CAMD
      GO TO 50                                                          CAMD
  350 IERCDE = 15                                                       MOD2
      CALL ERRPRO                                                       MOD2
      CALL SEGPRO(IND,1)                                                MOD2
      IF(IEATER .NE. 0)GO TO 500                                        MOD1
      IF(IND .EQ. 0)GO TO 350                                           MOD1
      BACKSPACE ICAMER                                                  MOD1
      GO TO 500                                                         MOD1
  425 IERCDE = 2                                                        CAMD
      CALL ERRPRO                                                       CAMD
      GO TO 500                                                         CAMD
  450 ICAMER = -1                                                       CAMD
      WRITE(ICAME)IZ,(IZAP,I=1,49)                                      CAMD
      REWIND ICAME                                                      CAMD
  500 RETURN                                                            CAMD
  900 FORMAT(1H0,49X,11HCAMERR FILE/3X, 8HICTRY = ,A4, 8H IREG = ,I3,   CAMD
     1 9H IZONE = ,I5,10H ISTRAT = ,I4, 8H ISUB = ,I4, 8H ISEG = ,I4/   CAMD
     23X,75H(PWW(I),PWM(I),PWO(I),BWW(I),SIGWW(I),BWM(I),SIGWM(I),BWO(I)MOD3
     2,SIGWO(I),BPW(I),SIGPW(I),I=1,4) = ,4(/3X,F4.2,1X,F4.2,1X,F4.2,1X,MOD1
     3F6.3,1X,F6.3,1X,F6.3,1X,F4.2,1X,F4.2,1X,F4.2,1X,F6.3,1X,F4.2))    MOD1
      END                                                               CAMD
```

```
@ ELT ERRPRO,1,760402, 50775   , 1

      SUBROUTINE ERRPRO                                                 ERRPRO
C     THIS ROUTINE CONTROLS THE PROCESSING OF ALL ERRORS,PRINTS ERROR   ERRPRO
C     MESSAGES AND CARD IMAGES ACCORDING TO ERROR TYPE                  ERRPRO
      COMMON /FLAG/                                                     FLAG
     1ICTRSV,IREGSV,IZONSV,INP,IOUT,ICASE,ILIST,IDFRS(2),IOTUS(2),      FLAG
     2IYESU,ISIGEX,ICAMER,ICSFSH,ICTRY,IREG,IZONE,ISTRAT,ISUB,IREGP,    FLAG
     3IZONEP,ISTRAP,IFATER,IZSGC,ITYPE,IFATEY,IFATEC,IFATES,IRANGE,     FLAG
     4ICAMR,IYESR,ISIGR,IERR,IPRINT,IFFSU,ISBUS,IERCDE,ISUBCT,          FLAG
     5IYSFCT,ICMCT,ISGCT,IYSERC,ICMERC,ISIGEC,IPSEG,IYESER,ICAME,       FLAG
     6ISIGE,IREG2P,IZON2P,ISTR2P,ISUB2P,ISEG2P,IREG1P,IZON1P,ISTR1P,    FLAG
     7IREG3P,IZON3P,ILEVEL,IFILT1,IFILT2                                FLAG
     8,ILEVES                                                           MOD3
      COMMON /CAMSIN/                                                   CAMSIN
     1ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,PWW(4),PWM(4),PWO(4),      CAMSIN
     2BWW(4),BWM(4),BWO(4),SIGWW(4),SIGWM(4),SIGWO(4),BPW(4),SIGPW(4)   CAMSIN
      COMMON /SIGIN/                                                    SIGIN
     1ICTRY3,IREG3,IZONE3,B1W,B2W,SIG1W(6),SIG2W(6),B1M,B2M,SIG1M(6),   SIGIN
     2SIG2M(6),B10,B20,SIG10(6),SIG20(6),B1,B2,SIG1(6),SIG2(6)          SIGIN
      COMMON /YESIN/                                                    YESIN
     1ICTRY1,IREG1,IZONE1,ISTRA1,YIELDI,IERDTE(3,6),BIASYI(6),SIGYII(6),YESIN
     2YIELD,IRDTE(6),BIASY(6),SIGYIE(6)                                 YESIN
      DATA IZ/4HZZZZ/,IZER/0/,ICM/6HCAMERR/,IYS/6HYESERR/,ISG/6HSIGEXT/,ERRPRO
     1I1,I2,I3,I4,I5,I6,I7/1,2,3,4,5,6,7/                               ERRPRO
      IF(ITYPE .EQ. 1)IERRP = IYS                                       MOD2
      IF(ITYPE .EQ. 2)IERRP = ICM                                       MOD2
      IF(ITYPE .EQ. 3)IERRP = ISG                                       MOD2
      IF(IERCDE .GT. 19)GO TO 5                                         ERRPRO
      IF(IERCDE .LT. 10)GO TO 300                                       ERRPRO
      IF(IERCDE .GE. 15 .AND. IERCDE .LE. 19)GO TO 305                  ERRPRO
    5 IF(IRANGE .EQ. 0)GO TO 50                                         ERRPRO
      IF(ITYPE .EQ. 1)GO TO 20                                          ERRPRO
      IF(ITYPE .EQ. 2)GO TO 10                                          ERRPRO
      IFATES = 1                                                        ERRPRO
      WRITE(ISIGE)IZ,(IZER,I=1,58)                                      ERRPRO
      REWIND ISIGE                                                      ERRPRO
      GO TO 45                                                          ERRPRO
   10 IFATEC = 1                                                        ERRPRO
      WRITE(ICAME)IZ,(IZER,I=1,49)                                      ERRPRO
      REWIND ICAME                                                      ERRPRO
      GO TO 45                                                          ERRPRO
   20 CONTINUE                                                          MOD2
      IFATEY = 1                                                        ERRPRO
      WRITE(IYESER)IZ,(IZER,I=1,22)                                     ERRPRO
      REWIND IYESER                                                     ERRPRO
   45 WRITE(IOUT,900)IERRP                                              ERRPRO
   50 IF(IERCDE .GT.19)GO TO 100                                        ERRPRO
      IF(ITYPE .NE. 3)GO TO 55                                          ERRPRO
      WRITE(IOUT,901)ICTRSV,IREGSV,IZONSV                               ERRPRO
      GO TO 60                                                          ERRPRO
   55 WRITE(IOUT,902)ICTRY,IREG,IZONE,ISTRAT,ISUB                       ERRPRO
   60 GO TO 350                                                         ERRPRO
  100 IF(IPRINT .NE. 0)GO TO 350                                        ERRPRO
      IPRINT =1                                                         ERRPRO
      IF(ILIST .NE. 0)GO TO 350                                         ERRPRO
      IF(ITYPE .EQ. 1)GO TO 120                                         ERRPRO
      IF(ITYPE .EQ. 2)GO TO 110                                         ERRPRO
      WRITE(IOUT,903)                                                   ERRPRO
      WRITE(IOUT,904)ICTRY3,IREG3,IZONE3,B1W,B2W,(SIG1W(I),I=1,6),(SIG2WERRPRO
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
     1(I),I=1,6),I1,I7                                                    ERRPRO
      WRITE(IOUT,905)B10,B20,(SIG10(I),I=1,6),(SIG20(I),I=1,6),I3,I7      ERRPRO
      WRITE(IOUT,905)B1,B2,(SIG1(I),I=1,6),(SIG2(I),I=1,6),I4,I7          ERRPRO
      GO TO 350                                                           ERRPRO
  110 WRITE(IOUT,906)                                                     ERRPRO
      WRITE(IOUT,907)ISEG2,ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,(PWM(I),      ERRPRO
     1I=1,4),(BWA(I),I=1,4),(SIGWW(I),I=1,4),I1,I6                        ERRPRO
      WRITE(IOUT,908)(PWM(I),I=1,4),(BWM(I),I=1,4),(SIGWM(I),I=1,4),I2,   ERRPRO
     1I6                                                                  ERRPRO
      WRITE(IOUT,908)(PWO(I),I=1,4),(BWO(I),I=1,4),(SIGWO(I),I=1,4),I3,   ERRPRO
     1I6                                                                  ERRPRO
      WRITE(IOUT,909)(SIGPW(I),I=1,4),(BPW(I),I=1,4),I4,I6                ERRPRO
      GO TO 350                                                           ERRPRO
  120 WRITE(IOUT,910)                                                     ERRPRO
      WRITE(IOUT,911)ICTRY1,IREG1,IZONE1,ISTRA1,YILLOT,((IEROTE(I,J),I=   ERRPRO
     11,3),J=1,6),I1,I5                                                   ERRPRO
      WRITE(IOUT,912)(BIASYI(I),I=1,6),(SIGYII(I),I=1,6),I2,I5            ERRPRO
      GO TO 350                                                           ERRPRO
  300 IFATER = 1                                                          ERRPRO
      IYESU =-1                                                           ERRPRO
      ICAMER =-1                                                          ERRPRO
      ISIGEX =-1                                                          ERRPRO
      WRITE(IYESER)IZ,(IZER,I=1,22)                                       ERRPRO
      WRITE(ICAME)IZ,(IZER,I=1,49)                                        ERRPRO
      WRITE(ISIGE)IZ,(IZER,I=1,58)                                        ERRPRO
      REWIND IYESER                                                       MOD1
      REWIND ISIGE                                                        ERRPRO
      REWIND ICAME                                                        ERRPRO
      IF(IERCDE .NE. 2)GO TO 350                                          ERRPRO
  305 IF(ITYPE .EQ. 1)GO TO 320                                           ERRPRO
      IF(ITYPE .EQ. 2)GO TO 310                                           ERRPRO
      WRITE(IOUT,901)ICTRY3,IREG3,IZONE3                                  ERRPRO
      GO TO 350                                                           ERRPRO
  310 WRITE(IOUT,913)ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2               ERRPRO
      GO TO 350                                                           ERRPRO
  320 WRITE(IOUT,914)ICTRY1,IREG1,IZONE1,ISTRA1                           ERRPRO
  350 GO TO(400,405,410,500,500,500,500,500,500,415,420,500,500,500,425,ERRPRO
     1500,500,500,500,430,435,440,445,450,455,460,465,470,475), IERCDE    MOD3
  400 WRITE(IOUT,915)                                                     ERRPRO
      GO TO 600                                                           ERRPRO
  405 WRITE(IOUT,916)IERRP                                                ERRPRO
      GO TO 600                                                           ERRPRO
  410 WRITE(IOUT,917)                                                     ERRPRO
      GO TO 600                                                           ERRPRO
  415 WRITE(IOUT,918)IERRP                                                ERRPRO
      GO TO 600                                                           ERRPRO
  420 WRITE(IOUT,919)IPSEG                                                ERRPRO
      GO TO 600                                                           ERRPRO
  425 WRITE(IOUT,920)IERRP                                                ERRPRO
      GO TO 600                                                           ERRPRO
  430 WRITE(IOUT,921)                                                     ERRPRO
      GO TO 600                                                           ERRPRO
  435 WRITE(IOUT,922)                                                     ERRPRO
      GO TO 600                                                           ERRPRO
  440 WRITE(IOUT,923)                                                     ERRPRO
      GO TO 600                                                           ERRPRO
  445 WRITE(IOUT,924)                                                     ERRPRO
      GO TO 600                                                           ERRPRO
  450 WRITE(IOUT,925)                                                     ERRPRO
      GO TO 600                                                           ERRPRO
```

```
  455 WRITE(IOUT,926)                                                   ERRPRO
      GO TO 600                                                         ERRPRO
  460 WRITE(IOUT,927)                                                   ERRPRO
      GO TO 600                                                         ERRPRO
  465 WRITE(IOUT,928)                                                   ERRPRO
      GO TO 600                                                         ERRPRO
  470 WRITE(IOUT,929)                                                   ERRPRO
      GO TO 600                                                         ERRPRO
  475 WRITE(IOUT,931)                                                   MOD3
      GO TO 600                                                         MOD3
  500 WRITE(IOUT,930)                                                   ERRPRO
      STOP                                                              ERRPRO
  600 IERR = IERR + 1                                                   ERRPRO
      RETURN                                                            ERRPRO
  900 FORMAT(1H0,20X,25HLEVEL 2 ERROR,WRITING OF ,A6,13H FILE STOPPED)  ERRPRO
  901 FORMAT(1H0,2X,6HICTRY ,A4,6H IREG ,I3,7H IZONE ,I3)               ERRPRO
  902 FORMAT(1H0,2X,6HICTRY ,A4,6H IREG ,I3,7H IZONE ,I3,7H ISTRAT ,I4, ERRPRO
     16H ISUB ,I4)                                                      ERRPRO
  903 FORMAT(1H0,45X,18HSTG EXT DATA GROUP)                             ERRPRO
  904 FORMAT(1H ,14X,A4,I3,I3,2F6.3,12F4.2,4X,2I1)                      ERRPRO
  905 FORMAT(1H ,24X,2F6.3,12F4.2,4X,2I1)                               ERRPRO
  906 FORMAT(1H0,47X,15HCAMS DATA GROUP)                                ERRPRO
  907 FORMAT(1H ,14X,I4,A4,I3,I3,I4,I4,4F4.2,4F6.3,4F4.2,2I1)           ERRPRO
  908 FORMAT(1H ,36X,4F4.2,4F6.3,4F4.2,2I1)                             ERRPRO
  909 FORMAT(1H ,36X,4F4.2,4F6.3,16X,2I1)                               ERRPRO
  910 FORMAT(1H0,47X,14HYES DATA GROUP)                                 ERRPRO
  911 FORMAT(1H ,14X,A4,I3,I3,I4,F5.2,18I2,19X,2I1)                     ERRPRO
  912 FORMAT(1H ,28X,6F5.1,6F5.2,2I1)                                   MOD3
  913 FORMAT(1H ,2X,6HICTRY ,A4,6H IREG ,I3,7H IZONE ,I3,7H ISTRAT ,I4, ERRPRO
     16H ISUB ,I4,6H ISEG ,I4)                                          ERRPRO
  914 FORMAT(1H ,2X,6HICTRY ,A4,6H IREG ,I3,7H IZONE ,I3,7H ISTRAT ,I4) ERRPRO
  915 FORMAT(1H0,107H*** EITHER THE SUBSTRATA HIST. FILE HAS NOT BEEN MOERRPRO
     1UNTED OR IT HAS THE WRONG CASE NO. PROCESSING IS STOPPED)         ERRPRO
  916 FORMAT(1H0, 8H*** THE ,A6,57H INPUT DATA SET IS OUT OF ORDER. PROCERRPRO
     1ESSING IS TERMINATED)                                             ERRPRO
  917 FORMAT(1H0,44H*** ID IN IDIOS IS NOT GREATER OR = TO IDFRS)       ERRPRO
  918 FORMAT(1H0,4H*** ,A6,69H INPUT DATA SET DOES NOT HAVE A RECORD WHIERRPRO
     1CH CORRESPONDS TO SUBHST ID)                                      ERRPRO
  919 FORMAT(1H0,52H*** CAMS INPUT DATA GROUP DOES NOT EXIST FOR SEGMENTERRPRO
     1 ,I4,17H IN SUBHST RECORD)                                        ERRPRO
  920 FORMAT(1H0, 8H*** THE ,A6,67H INPUT DATA GROUP DOES NOT MATCH WITHERRPRO
     1 SUBHST FILE AND IS EXTRA DATA)                                   ERRPRO
  921 FORMAT(1H0,49H*** IFRDIF ARRAY HAS DATES NOT IN ASCENDING ORDER)  ERRPRO
  922 FORMAT(1H0,36H*** YIELDI IS NOT BETWEEN 0 AND 99.9)               ERRPRO
  923 FORMAT(1H0,50H*** ABSOLUTE VALUE OF BIASYI IS NOT LESS THAN 99.9) ERRPRO
  924 FORMAT(1H0,36H*** SIGYII IS NOT BETWEEN 0 AND 99.9)               ERRPRO
  925 FORMAT(1H0,30H*** PW- IS NOT BETWEEN 0 AND 1)                     ERRPRO
  926 FORMAT(1H0,54H*** ABSOLUTE VALUE OF BW- OR BPW IS NOT LESS THAN 9.ERRPRO
     199)                                                               ERRPRO
  927 FORMAT(1H0,44H*** SIGW- OR SIGPW IS NOT BETWEEN 0 AND 9.99)       ERRPRO
  928 FORMAT(1H0,54H*** ABSOLUTE VALUE OF B1- OR B2- IS NOT LESS THAN 9.ERRPRO
     199)                                                               ERRPRO
  929 FORMAT(1H0,44H*** SIG1- OR SIG2- IS NOT BETWEEN 0 AND 9.99)       ERRPRO
  930 FORMAT(1H0,33H*** BAD ERROR CODE. JOB ABANDONED)                  ERRPRO
  931 FORMAT(1H0,51H*** IFRDIF ARRAY HAS A BAD YEAR,MONTH OR DAY NUMBER)MOD3
      END                                                               ERRPRO
```

```
 • FLT LFPA,1,760402, 58777     , 1

      SUBROUTINE LFPA(FLDA,LMO,LYR,DAYS)                              LFPA
C                                                                     LFPA
C MODULE - P1LFPA                                                     LFPA
C AUGUST 27, 1975                                                     LFPA
C                                                                     LFPA
C                                                                     LFPA
C  JUNE 26,1973                                                       LFPA
C                                                                     LFPA
C  MODULE PPLFPA - POINT TARGET PERFORMANCE PREDICTOR                 LFPA
C                                                                     LFPA
C  GIVEN  DAY,MONTH,YEAR - SUBR. LFPA RETURNS THE RIGHT ASCENSION OF  LFPA
C  GREENWICH AT MIDNIGHT OF A GIVEN DAY                               LFPA
C                                                                     LFPA
C     REFERENCE EPOCH IS 0 HOUR 1JAN1950                              LFPA
C                                                                     LFPA
C     COMPUTE DAYS IN FULL YEARS FROM EPOCH TO LYR                    LFPA
C     DAYS TO 1JAN1963 IS 4748.                                       LFPA
C                                                                     LFPA
      DAYS=4748.                                                      LFPA
C     NOTE--LYR MUST BE GREATER THAN OR EQUAL TO (19)64               LFPA
      LASTYR=LYR-1                                                    LFPA
      DO 20 I=63,LASTYR                                               LFPA
      KREMD=MOD((I-60),4)                                             LFPA
      IF(KREMD.GT.0) GO TO 10                                         LFPA
      DAYS=DAYS+366                                                   LFPA
      GO TO 20                                                        LFPA
   10 DAYS=DAYS+365                                                   LFPA
   20 CONTINUE                                                        LFPA
      IF(LMO-1)30,40,30                                               LFPA
   30 IF(LMO-2)60,50,60                                               LFPA
   40 DAYS=DAYS+FLDA-1.                                               LFPA
      GO TO 270                                                       LFPA
   50 DAYS=DAYS+FLDA+30.                                              LFPA
      GO TO 270                                                       LFPA
   60 KDEL=MOD((LYR-60),4)                                            LFPA
      IF(KDEL.GT.0) GO TO 70                                          LFPA
      DAYS=DAYS+59.                                                   LFPA
      GO TO 80                                                        LFPA
   70 DAYS=DAYS+58.                                                   LFPA
   80 IF(LMO-3) 270,170,90                                            LFPA
   90 IF(LMO-5) 180,190,100                                           LFPA
  100 IF(LMO-7) 200,210,110                                           LFPA
  110 IF(LMO-9) 220,230,120                                           LFPA
  120 IF(LMO-11)240,250,260                                           LFPA
  170 DAYS=DAYS+FLDA                                                  LFPA
      GO TO 270                                                       LFPA
  180 DAYS=DAYS+FLDA+31.                                              LFPA
      GO TO 270                                                       LFPA
  190 DAYS=DAYS+FLDA+61.                                              LFPA
      GO TO 270                                                       LFPA
  200 DAYS=DAYS+FLDA+92.                                              LFPA
      GO TO 270                                                       LFPA
  210 DAYS=DAYS+FLDA+122.                                             LFPA
      GO TO 270                                                       LFPA
  220 DAYS=DAYS+FLDA+153.                                             LFPA
      GO TO 270                                                       LFPA
  230 DAYS=DAYS+FLDA+184.                                             LFPA
      GO TO 270                                                       LFPA
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
  240 DAYS=DAYS+FIDA+214.                                          LFPA
      GO TO 270                                                    LFPA
  250 DAYS=DAYS+FIDA+245.                                          LFPA
      GO TO 270                                                    LFPA
  260 DAYS=DAYS+FIDA+275.                                          LFPA
  270 CONTINUE                                                     LFPA
C                                                                  LFPA
      RETURN                                                       LFPA
C                                                                  LFPA
      END                                                          LFPA
```

```
 FLT SEF,1,760426, 48475       , 1

     C        PROGRAM SFE(ITEST=401,OTEST=401,TAPE5=ITEST,TAPE6=OTEST,SIGEXT=100SFE
     C       12,SUBHST=1002,YESERR=1002,CAMERR=1002,SIGECD=401,CAMECD=401,YESECDSFE
     C       2=401,TAPE8=CAMERR,TAPE9=SIGEXT,TAPE11=YESERR,TAPE3=SUBHST,TAPE1=  MOD1
     C       3YESECD,TAPE2=SIGECD,TAPE4=CAMECD)                                 SFE
              COMMON /FLAG/                                                     FLAG
             1ICTRSV,IREGSV,IZONSV,INP,IOUT,ICASE,ILIST,IDFRS(2),IDTUS(2),      FLAG
             2IYESU,ISIGEX,ICAMER,ICSESH,ICTRY,IREG,IZONE,ISTRAT,ISUB,IREGP,    FLAG
             3IZONEP,ISTRAP,IFATER,IZSGC,ITYPE,IFATEY,IFATEC,IFATES,IRANGE,     FLAG
             4ICAMR,IYESR,ISIGR,IERR,IPRINT,IFFSB,ISUBS,IFHCUF,ISUBCT,          FLAG
             5IYSCT,ICMCT,ISGCT,IYSERC,ICMERC,ISIGEC,IPSEG,IYESER,ICAME,        FLAG
             6ISIGE,IREG2P,IZON2P,ISTR2P,ISUB2P,ISEG2P,IREG1P,IZON1P,ISTR1P,    FLAG
             7IREG3P,IZON3P,ILEVEL,IFILT1,IFILT2                                FLAG
             8,ILEVES                                                           MOD3
              COMMON /SUBHIS/                                                   SUBHIS
             1NSEG,IDSEG(150),IGRP,PWHTH,AREA,PHIHT,NAGR,NAL,DELPW,DELPM,       SUBHIS
             2CV(4),NAME(2),ICSE,INXSEG                                         SUBHIS
              COMMON /CAMSIN/                                                   CAMSIN
             1ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,PWW(4),PWM(4),PWO(4),      CAMSIN
             2BWW(4),BWM(4),BWO(4),SIGWW(4),SIGWM(4),SIGWO(4),BPW(4),SIGPW(4)   CAMSIN
              COMMON /SIGIN/                                                    SIGIN
             1ICTRY3,IREG3,IZONE3,B1W,B2W,SIG1W(6),SIG2W(6),B1M,B2M,SIG1M(6),   SIGIN
             2SIG2M(6),B10,B20,SIG10(6),SIG20(6),B1,B2,SIG1(6),SIG2(6)          SIGIN
              COMMON /YESIN/                                                    YESIN
             1ICTRY1,IREG1,IZONE1,ISTRA1,YIELD1,IERDIF(3,6),BIASYI(6),SIGYII(6),YESIN
             2YIELD,IRDIF(6),BIASY(6),SIGYIL(6)                                 YESIN
              DIMENSION NAMI(2),NAMCAM(2),NAMYES(2),NAMSIG(2),IDAT(4),IDAT1(3)  SFE
              DATA NAMCAM/4HCAMS,4H ERR/,NAMYES/4HYESE,4HRROR/,NAMSIG/4HSIGE,   SFE
             14HXTEN/,IDAT/6H       ,6HYESERR,6HCAMERR,6HSIGEXT/                SFE
              DATA IPAD/07777777777/,IZER/0/,NAMI/4HSUB ,4HHIST/                          *NEW
              IFATER = 0                                                        SFE       **-1
              IFATEY = 0                                                        SFE
              IFATEC = 0                                                        SFE
              ISIGR = 0                                                         SFE
              IFATES = 0                                                        SFE
              ICAMR = 0                                                         SFE
              IYESR = 0                                                         SFE
              IFILT1 = 0                                                        SFE
              IFILT2 = 0                                                        SFE
              IPRINT = 0                                                        SFE
              IFFSB = 0                                                         SFE
              ITYPE = 0                                                         SFE
              IERR = 0                                                          SFE
              IZSGC = 0                                                         SFE
              ISUBCT = 0                                                        SFE
              IYSCT = 0                                                         SFE
              ICMCT = 0                                                         SFE
              ISGCT = 0                                                         SFE
              IYSERC = 0                                                        SFE
              ICMERC = 0                                                        SFE
              ISIGEC = 0                                                        SFE
              IREGP = 0                                                         SFE
              IZONEP = 0                                                        SFE
              ISTRAP = 0                                                        SFE
              IREG2P = 0                                                        SFE
              IZON2P = 0                                                        SFE
              ISTR2P = 0                                                        SFE
              ISUB2P = 0                                                        SFE
              ISEG2P = 0                                                        SFE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IREGIP = 0                                                        SFE
      IZONIP = 0                                                        SFE
      ISIPIP = 0                                                        SFE
      IREG3P = 0                                                        SFE
      IZONSP = 0                                                        SFE
      READ(INP,900)ICASE,ILIST,(IDFRS(I),I=1,2),IDTOS(1),IDTOS(2),IYT,  SFE
     1IST,ICTE,ICSESH                                                   SFE
      WRITE(IOUT,901)ICASE,ILIST,(IDFRS(I),I=1,2),IDTOS(1),IDTOS(2),IYT,SFE
     1IST,ICTE,ICSESH                                                   SFE
      IF(IYT .NE. 0)IYESU = IYT                                         SFE
      IF(IST .NE. 0)ISIGEX = IST                                        SFE
      IF(ICTE .NE. 0)ICAMER = ICTE                                      SFE
      IF(IDFRS(1) .NE. 0 .OR. IDTOS(1) .NE. 0)GO TO 10                  SFE
      IDTOS(1) = IPAD                                                   SFE
      IDTOS(2) = IPAD                                                   SFE
   10 IF(IDTOS(1) .LT. IDFRS(1))GO TO 20                                SFE
      IF(IDTOS(1) .NE. IDFRS(1))GO TO 25                                            *NEW
      IF(IDTOS(2) .LT. IDFRS(2))GO TO 20                                SFE
      GO TO 25                                                          SFE
   20 IERCDE = 3                                                        SFE
      CALL ERRPRO                                                       SFE
      GO TO 50                                                          SFE
   25 REWIND ISPHS                                                      SFE
      IF(IYESU .GT. 0)REWIND IYESU                                      MOD2
      IF(ISIGEX .GT. 0)REWIND ISIGEX                                    MOD2
      IF(ICAMER .GT. 0)REWIND ICAMER                                    MOD2
      READ(ISPHS)NAME ,      ICSE,  IMXSEG                              SFE
      IF(ICSESH .EQ. 0)GO TO 27                                         SFE
      IF(ICSESH.NE. ICSE  )GO TO 30                                     SFE
   27 IF(NAME(1).NE.NAMT(1) .OR.NAME(2).NE. NAMT(2))GO TO 30            SFE
      GO TO 35                                                          SFE
   30 IERCDE = 1                                                        SFE
      CALL ERRPRO                                                       SFE
      GO TO 50                                                          SFE
   35 IF(IYESU .GT. 0)WRITE(IYESER)NAMYES,ICASE,(IZER,I=1,20)           MOD1
      IF(ISIGEX .GT. 0)WRITE(ISIGE)NAMSIG,ICASE,(IZER,I=1,56)           MOD1
      IF(ICAMER .GT. 0)WRITE(ICAME)NAMCAM,ICASE,(IZER,I=1,47)           MOD1
      CALL SEECON                                                       SFE
      WRITE(IOUT,902)ISUBCT,IYSCT,ICMCT,ISGCT,IYSERC,ICASE,ICMERC,ICASE,SFE
     1ISIGEF,ICASE,IERR                                                 SFE
   50 IF(IFATER .NE. 0)WRITE(IOUT,903)                                  SFE
      IF(IFATER .EQ. 0)WRITE(IOUT,904)                                  SFE
      IF(IFATEY .NE. 0)GO TO 60                                         SFE
      IF(IFATEC .NE. 0)GO TO 60                                         SFE
      IF(IFATES .NE. 0)GO TO 60                                         SFE
   55 STOP                                                              SFE
   60 DO 65 I=1,3                                                       SFE
      IDATI(I) = IDAT(I)                                                SFE
   65 CONTINUE                                                          SFE
      IF(IFATEY .NE. 0)IDATI(1) = IDAT(2)                               SFE
      IF(IFATEC .NE. 0)IDATI(2) = IDAT(3)                               SFE
      IF(IFATES .NE. 0)IDATI(3)= IDAT(4)                                SFE
      WRITE(IOUT,905)(IDATI(I),I=1,3)                                   SFE
      GO TO 55                                                          SFE
  900 FORMAT(I4,I1,4I3,3I2,I4)                                          SFE
  901 FORMAT(1H1,2X,16A,16HIDFRS       IDTOS/3X, 64HICASE  ILIST  REG  ZOMOD1
     1NE  REG  ZONE  IYESU  ISIGEX  ICAMR  ICSESH/4X,I4,3X,I2,4X,I3,3X, MOD1
     1I3,2X,I3,3X,I3,3X,I2,6X,I2,5X,I2,5X,I4)                           SFE
  902 FORMAT(1H1,44X,22HSEE STATUS INFORMATION/                         SFE
     13X,I4,39H SUBSTRATA HISTORICAL FILE RECORDS READ/                 SFE
```

```
     23X,I4,24H YES DATA CARD SETS READ/                                MOD3
     33X,I4,31H CAMS ERROR DATA CARD SETS READ/                         MOD3
     43X,I4,40H SIGNATURE EXPENSION DATA CAPD SETS READ/                MOD3
     43X,I4,42H RECORDS WRITTEN ON YESERR FILE. ICSEYM = ,I4/           SFE
     53X,I4,42H RECORDS WRITTEN ON CAMERR FILE. ICSECE = ,I4/           SFE
     63X,I4,42H RECORDS WRITTEN ON SIGEXT FILE. ICSESF = ,I4/           SFE
     73X,I4,22H INPUT ERRORS DETECTED)                                  SFE
 903 FORMAT(1H1,2X,17HJOB IS INCOMPLETE)                                SFE
 904 FORMAT(1H1,2X,13HJOB COMPLETED)                                    SFE
 905 FORMAT(1H0,2X,64HFATAL ERROR OCCURRED IN SEE. FOLLOWING FILES MAY  SFE
    1NOT BE ANY GOOD/14X,3(A6,2X))                                      SFE
      END                                                               SFE
```

* FLT SEECON,1,760426, 48476   , 1

```
      SUBROUTINE SEECON                                                 SEECON
C     READS THE SUBHST FILE AND CONTROLS THE READING AND PROCESSING OF  SEECON
C     THE 3 INPUT FILES.                                                SEECON
      COMMON /FLAG/                                                     FLAG
     1ICTRSV,IREGS,,IZONSV,INP,IOUT,ICASE,ILIST,IDFRS(2),IDTOS(2),      FLAG
     2IYESU,ISIGEX,ICAMER,ICSFSH,ICTRY,IREG,IZONE,ISTRAT,ISUB,IREGP,    FLAG
     3IZONEP,ISTRAP,IEATER,IZSGC,ITYPE,IEATEY,IEATEC,IEATES,IRANGE,     FLAG
     4ICAMR,IYESR,ISIGR,IERR,IPRINT,IFFSB,ISUHS,IERCDF,ISUBCT,          FLAG
     5IYSCT,ICMCT,ISGCT,IYSFRC,ICMFRC,ISIGEC,IPSEC,IYESFR,ICAME,        FLAG
     6ISIGE,IREG2P,IZON2P,ISTR2P,ISUB2P,ISEG2P,IREG1P,IZON1P,ISTR1P,    FLAG
     7IREG3P,IZON3P,ILEVEL,IFILT1,IFILT2                                FLAG
     8,ILEVLS                                                           MOD3
      COMMON /SIGIN/                                                    SIGIN
     1ICTRY3,IREG3,IZONE3,B1W,B2W,SIG1W(6),SIG2W(6),B1M,B2M,SIG1M(6),   SIGIN
     2SIG2M(6),B1D,B2D,SIG1D(6),SIG2D(6),B1,B2,SIG1(6),SIG2(6)         SIGIN
      COMMON /YESIN/                                                    YESIN
     1ICTRY1,IREG1,IZONE1,ISTRA1,YIELD1,IERDIF(3,6),BIASY1(6),SIGY1I(6),YESIN
     2YIELD,IRDIF(6),BIASY(6),SIGYIE(6)                                YESIN
      COMMON /CAMSIN/                                                   CAMSIN
     1ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,PWW(4),PWM(4),PWD(4),      CAMSIN
     2BWW(4),BWM(4),BWD(4),SIGWW(4),SIGWM(4),SIGWD(4),BP(4),SIGPW(4)    CAMSIN
      COMMON /SUBHIS/                                                   SUBHIS
     1NSEG,IDSEG(150),IGRP,PWWTH,AREA,PWTHT,NAGR,NAL,DELPW,DELPM,       SUBHIS
     2CV(4),NAME(2),ICSE,IMXSEG                                         SUBHIS
      DATA I//0027Z//,IPAD/  07777777777/                               SEECON
      IFIRST = 1                                                        SEECON
   10 IF(IEFSB .NE. 0)GO TO 35                                          SEECON
      READ(ISUHS)ICTRY,IREG,IZONE,ISTRAT,ISUB,NSEG,(IDSEG(I),I=1,IMXSEG)SEECON
     1,IGRP,PWWTH,AREA,PWTHT,NAGR,NAL,DELPW,DELPM,(CV(I),I=1,4)         SEECON
      IF(ICTRY .NE. I7)GO TO 20                                         SEECON
      IEFSB = 1                                                         SEECON
      IREG = IPAD                                                       SEECON
      IZONE = IPAD                                                      SEECON
      ISTRAT = IPAD                                                     SEECON
      ISUB = IPAD                                                       SEECON
      IRANGE = 0                                                        SEECON
      IZSGC = 0                                                         SEECON
      ILEVEL = 5                                                        MOD2
      GO TO 26                                                                  *NEW
   20 ISUBCT = ISUBCT + 1                                               SEECON  **-1
      ILEVEL = 1                                                        SEECON
      IF(ISTRAT .NE. ISTRAP)ILEVEL = 2                                  SEECON
      IF(IZONE .NE. IZONEP)ILEVEL = 3                                   SEECON
      IF(IREG .NE. IREGP)ILEVEL = 4                                     SEECON
   25 IREGP = IREG                                                      SEECON
      IZONEP = IZONE                                                    SEECON
      ISTRAP = ISTRAT                                                   SEECON
   26 IF(IREG .LT. IDFRS(1) .OR. IREG .GT. IDTOS(1))GO TO 30                    *NEW
      IF(IREG .EQ. IDFRS(1))GO TO 28                                            *NEW
      IF(IREG .EQ. IDTOS(1))GO TO 29                                            *NEW
      GO TO 27                                                                  *NEW
   28 IF(IZONE .LT. IDFRS(2))GO TO 30                                           *NEW
      GO TO 27                                                                  *NEW
   29 IF(IZONE .GT. IDTOS(2))GO TO 30                                           *NEW
   27 IRANGE = 1                                                                *NEW
      GO TO 35                                                          SEECON  **-3
   30 IRANGE = 0                                                        SEECON
   35 IF(IYESU .LT. 0)GO TO 45                                          SEECON
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      IF(ILEVEL .LE. 1)GO TO 45                                SFECON
      CALL YESD                                                SFECON
   45 IF(ICAMER .LT. 0)GO TO 50                                SFECON
      CALL CAMD                                                SFECON
   50 IF(ISIGFX .LT. 0)GO TO 65                                SFECON
      IF(ILEVEL .LE. 2)GO TO 65                                SFECON
      IF(IFIRST .EQ. 0)GO TO 55                                SFECON
      IFIRST = 0                                               SFECON
      GO TO 65                                                 SFECON
   55 CALL SIGD                                                SFECON
   65 IF(IYESU .GT. 0)GO TO 70                                 SFECON
      IF(ICAMER .GT. 0)GO TO 70                                SFECON
      IF(ISIGFX .GT. 0)GO TO 70                                SFECON
      RETURN                                                   SFECON
   70 IF(ILESW .EQ. 0)GO TO 75                                 SFECON
      IREGSV = IPAD                                            SFECON
      IZONSV = IPAD                                            SFECON
   75 IF(ILEVEL .GT. 2)GO TO 80                                SFECON
      IZSGC = IZSGC + NSEG                                     SFECON
      GO TO 10                                                 SFECON
   80 IZSGC = NSEG                                             SFECON
      ICTRSV = ICTRY                                           SFECON
      IREGSV = IREG                                            SFECON
      IZONSV =-IZONE                                           MOD3
      ILEVLS = ILEVEL                                          SFECON
      GO TO 10                                                 SFECON
      END                                                      SFECON
```

```
• FLT SEGPRO,1,760402, 58773   , 1

      SUBROUTINE SEGPRO(IND,IOP)                                        SEGPRO
C     READS 1 CAMS ERROR INPUT TO GROUP INTO CAMSIN COMMON              SEGPRO
      COMMON /FLAG/                                                     FLAG
     1ICTRSV,IREGSV,IZONSV,INP,IOUT,ICASE,ILIST,IDFRS(2),IDTOS(2),      FLAG
     2IYESU,ISIGEX,ICAMER,ICSESH,ICTRY,IREG,IZONE,ISTFAT,ISUB,IREGP,    FLAG
     3IZONEP,ISTRAP,IFATER,IZSGC,ITYPE,IFATEY,IFATEC,IFATES,IRANGE,     FLAG
     4ICAMR,IYESR,ISIGR,IFRP,IPRINT,IFFSB,ISUBS,IERCDF,ISUBCT,          FLAG
     5IYSCT,ICMCT,ISGCT,IYSERC,ICMERC,ISIGEC,IPSEG,IYESER,ICAML,        FLAG
     6ISIGL,IREG2P,IZON2P,ISTR2P,ISUB2P,ISEG2P,IREG1P,IZON1P,ISTR1P,    FLAG
     7IREG3P,IZON3P,ILEVEL,IFILT1,IFILT2                                FLAG
     8,ILEVES                                                           MOD3
      COMMON /CAMSIN/                                                   CAMSIN
     1ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,PWW(4),PWM(4),PWO(4),      CAMSIN
     2BWW(4),BWM(4),BWO(4),SIGWW(4),SIGWM(4),SIGWO(4),BPW(4),SIGPW(4)   CAMSIN
      EQUIVALENCE (PW,PWW),(BW,BWW),(SIGW,SIGWW)                        SEGPRO
      DIMENSION  PW(12),BW(12),SIGW(12)                                 SEGPRO
      DATA IZ/4HZZZZ/                                                   SEGPRO
      DIMENSION PWT(4),BWT(4),SIGWT(4)                                  MOD3
      IND = 0                                                           SEGPRO
      IPRINT = 0                                                        SEGPRO
      READ(ICAMER,900)ISEG2,ICTRY2,IREG2,IZONE2,ISTRA2,ISUB2,           SEGPRO
     1(PWT(I),I=1,4),(BWT(I),I=1,4),(SIGWT(I),I=1,4),ISEQ,ISET          MOD3
      IF(ICTRY2 .EQ. IZ)GO TO 310                                       SEGPRO
      IF(IREG2 .NE.IREG2P)GO TO 300                                     SEGPRO
      IF(IZONE2 .NE. IZON2P)GO TO 300                                   SEGPRO
      IF(ISTRA2 .NE. ISTR2P)GO TO 300                                   SEGPRO
      IF(ISUB2  .NE. ISUB2P)GO TO 300                                   SEGPRO
   10 DO 12 I=1,4                                                       MOD3
      PWW(I) = PWT(I)                                                   MOD3
      BWW(I) = BWT(I)                                                   MOD3
      SIGWW(I) = SIGWT(I)                                               MOD3
   12 CONTINUE                                                          MOD3
      IF(ILIST .NE. 0)WRITE(IOUT,901)ISEG2,ICTRY2,IREG2,IZONE2,ISTRA2,  MOD3
     1ISUB2,(PWW(I),I=1,4),(BWW(I),I=1,4),(SIGWW(I),I=1,4),ISEQ,ISET    MOD1
      IF(ISET .EQ. 6 .AND. ISEQ .EQ. 1)GO TO 20                         MOD1
   15 IERCDF = 2                                                        SEGPRO
      CALL ERPPRO                                                       SEGPRO
      GO TO 310                                                         SEGPRO
   20 READ(ICAMER,900)ISEGT,ICTRYT,IREGT,IZONET,ISTRT,ISUBT,            SEGPRO
     1(PWM(I),I=1,4),(BWM(I),I=1,4),(SIGWM(I),I=1,4),ISEQ,ISET          SEGPRO
      IF(ICTRYT .EQ. IZ)GO TO 15                                        SEGPRO
      IF(ILIST .NE. 0)WRITE(IOUT,902)(PWM(I),I=1,4),(BWM(I),I=1,4),     SEGPRO
     1(SIGWM(I),I=1,4),ISEQ,ISET                                        SEGPRO
      IF(IREGT .NE. IREG2 )GO TO 15                                     SEGPRO
      IF(IZONET .NE. IZONE2)GO TO 15                                    SEGPRO
      IF(ISTRT  .NE. ISTRA2)GO TO 15                                    SEGPRO
      IF(ISUBT  .NE. ISUB2 )GO TO 15                                    SEGPRO
      IF(ISEGT  .NE. ISEG2 )GO TO 15                                    SEGPRO
      IF(ISET .NE. 6 .OR. ISEQ .NE. 2)GO TO 15                          SEGPRO
      READ(ICAMER,900)ISEGT,ICTRYT,IREGT,IZONET,ISTRT,ISUBT,            SEGPRO
     1 (PWO(I),I=1,4),(BWO(I),I=1,4),(SIGWO(I),I=1,4),ISEQ,ISET         SEGPRO
      IF(ICTRYT .EQ. IZ)GO TO 15                                        SEGPRO
      IF(ILIST .NE. 0)WRITE(IOUT,902)(PWO(I),I=1,4),(BWO(I),I=1,4),     SEGPRO
     1(SIGWO(I),I=1,4),ISEQ,ISET                                        SEGPRO
      IF(IREGT .NE. IREG2)GO TO 15                                      SEGPRO
      IF(IZONET .NE. IZONE2)GO TO 15                                    SEGPRO
      IF(ISTRT .NE. ISTRA2)GO TO 15                                     SEGPRO
      IF(ISUBT .NE. ISUB2)GO TO 15                                      SEGPRO
```

```
      IF(ISEGT .NE. ISEG2)GO TO 15                                      SEGPRO
      IF(ISET .NE. 6 .OR. ISEQ .NE. 3)GO TO 15                          SEGPRO
      READ(ICAMER,903)ISEGT,ICTRYT,IRFGT,IZONET,ISTRT,ISUBT,            SEGPRO
     1(SIGPW(I),I=1,4),(BPW(I),I=1,4),ISEQ,ISET                         SEGPRO
      IF(ICTRYT .EQ. IZ)GO TO 15                                        SEGPRO
      IF(ILIST .NE. 0)WRITE(IOUT,904)(SIGPW(I),I=1,4),(BPW(I),I=1,4),   SEGPRO
     1ISEQ,ISET                                                         SEGPRO
      IF(IRFGT .NE. IREG2)GO TO 15                                      SEGPRO
      IF(IZONET.NE. IZONE2)GO TO 15                                     SEGPRO
      IF(ISTRT .NE. ISTRA2)GO TO 15                                     SEGPRO
      IF(ISUBT .NE. ISUB2)GO TO 15                                      SEGPRO
      IF(ISEGT .NE. ISEG2)GO TO 15                                      SEGPRO
      IF(ISET .NE. 6 .OR. ISEQ .NE. 4)GO TO 15                          SEGPRO
      ICNCT = ICNCT + 1                                                 SEGPRO
      DO 25 I=1,12                                                      SEGPRO
      IF(PW(I) .LT. 0  .OR. PW(I) .GT. 1.0)GO TO 30                     SEGPRO
   25 CONTINUE                                                          SEGPRO
      GO TO 35                                                          SEGPRO
   30 IERCOD = 24                                                       SEGPRO
      CALL ERRPRO                                                       SEGPRO
   35 DO 40 I=1,12                                                      SEGPRO
      IF(BW(I) .LE. -10.0 .OR.BW(I) .GE. 10.0)GO TO 50                  SEGPRO
   40 CONTINUE                                                          SEGPRO
      DO 45 I=1,4                                                       SEGPRO
      IF(BPW(I) .LE. -10.0 .OR. BPW(I) .GE. 10.0)GO TO 50               SEGPRO
   45 CONTINUE                                                          SEGPRO
      GO TO 55                                                          SEGPRO
   50 IERCOD = 25                                                       SEGPRO
      CALL ERRPRO                                                       SEGPRO
   55 DO 60 I=1,12                                                      SEGPRO
      IF(SIGW(I) .LT. 0 .OR. SIGW(I) .GE. 10.0)GO TO 70                 SEGPRO
   60 CONTINUE                                                          SEGPRO
      DO 65 I=1,4                                                       SEGPRO
      IF(SIGPW(I) .LT. 0 .OR. SIGPW(I) .GE. 10.0)GO TO 70               SEGPRO
   65 CONTINUE                                                          SEGPRO
      GO TO 320                                                         SEGPRO
   70 IERCOD = 26                                                       SEGPRO
      CALL ERRPRO                                                       SEGPRO
      GO TO 320                                                         SEGPRO
  300 IF(IOP .EQ. 0)GO TO 10                                            MOD1
  310 IND = 1                                                           SEGPRO
  320 RETURN                                                            SEGPRO
  900 FORMAT(I4,A4,2I3,2I4,4F4.2,4F6.3,4F4.2,2I1)                       SEGPRO
  901 FORMAT(1H0,47X,15HCARD DATA GROUP/1H ,14X,I4,A4,2I3,2I4,4F4.2,    SEGPRO
     14F6.3,4F4.2,2I1)                                                  SEGPRO
  902 FORMAT(1H ,36X,4F4.2,4F6.3,4F4.2,2I1)                             SEGPRO
  903 FORMAT(I4,A4,2I3,2I4,4F4.2,4F6.3,16X,2I1)                         SEGPRO
  904 FORMAT(1H ,36X,4F4.2,4F6.3,16X,2I1)                               SEGPRO
      END                                                               SEGPRO
```

```
# ELT SIGD,1,760402, 58768      , 1

      SUBROUTINE SIGD                                                   SIGD
C     THIS ROUTINE READS A SIGNATURE EXTENSION TO GROUP AND CHECKS FOR  SIGD
C     ERRORS AND WRITES THE RECORD ON THE CAMERR FILE                   SIGD
      COMMON /FLAG/                                                     FLAG
     1ICTRSV,IREGSV,IZONSV,INP,IOUT,ICASE,ILIST,IDFRS(2),IDTUS(2),      FLAG
     2IYESU,ISIGEX,ICAMER,ICSFSH,ICTRY,IREG,IZONE,ISTRAT,ISUB,IREGP,    FLAG
     3IZONEP,ISTRAP,IFATER,IZSGC,ITYPE,IFATEY,IFATEC,IFATES,IRANGE,     FLAG
     4ICAMR,IYESR,ISIGR,IERR,IPRINT,IEFSB,ISBHS,IERCOF,ISUBCT,          FLAG
     5IYSCT,ICMCT,ISGCT,IYSFRC,ICMLRC,ISIGEC,IPSEC,IYESER,ICAME,        FLAG
     6ISIGE,IREG2P,IZON2P,ISTR2P,ISUB2P,ISEG2P,IREG1P,IZON1P,ISTR1P,    FLAG
     7IREG3P,IZON3P,ILEVEL,IFILT1,IFILT2                                FLAG
     8,ILEVES                                                           MOD3
      COMMON /SIGIN/                                                    SIGIN
     1ICTRY3,IREG3,IZONE3,B1W,B2W,SIG1W(6),SIG2W(6),B1M,B2M,SIG1M(6),   SIGIN
     2SIG2M(6),B10,B20,SIG10(6),SIG20(6),B1,B2,SIG1(6),SIG2(6)          SIGIN
      DATA IZ/40077777/,IZER/0/                                         SIGD
      ITYPE = 3                                                         SIGD
      IPRINT = 0                                                        SIGD
      IF(ILEVES .GT. IFILT2)GO TO 50                                    MOD3
   10 IF(IFATES .NE. 0)GO TO 500                                        SIGD
      IF(ILIST .EQ. 2)WRITE(IOUT,900)ICTRSV,IREGSV,IZONSV,B1W,B2W,      SIGD
     1(SIG1W(I),I=1,6),(SIG2W(I),I=1,6),B1M,B2M,(SIG1M(I),I=1,6),       SIGD
     2(SIG2M(I),I=1,6),B10,B20,(SIG10(I),I=1,6),(SIG20(I),I=1,6),       SIGD
     3B1,B2,(SIG1(I),I=1,6),(SIG2(I),I=1,6)                             SIGD
      WRITE(ISIGE)ICTRSV,IREGSV,IZONSV,B1W,B2W,(SIG1W(I),I=1,6),        SIGD
     1(SIG2W(I),I=1,6),B1M,B2M,(SIG1M(I),I=1,6),(SIG2M(I),I=1,6),B10,B20SIGD
     2,(SIG10(I),I=1,6),(SIG20(I),I=1,6),B1,B2,(SIG1(I),I=1,6),         SIGD
     3(SIG2(I),I=1,6)                                                   SIGD
      ISIGEC = ISIGEC + 1                                               SIGD
      GO TO 500                                                         SIGD
   50 IF(ISIGR .EQ. 0)GO TO 65                                          SIGD
      ISIGR = 0                                                         SIGD
      GO TO 400                                                         SIGD
   65 READ(ISIGEX,901)ICTRY3,IREG3,IZONE3,B1W,B2W,(SIG1W(I),I=1,6),     SIGD
     1(SIG2W(I),I=1,6),ISEQ,ISET                                        SIGD
      IF(ICTRY3 .EQ. IZ)GO TO 475                                       SIGD
      IF(ILIST .NE. 0)WRITE(IOUT,902)ICTRY3,IREG3,IZONE3,B1W,B2W,       SIGD
     1(SIG1W(I),I=1,6),(SIG2W(I),I=1,6),ISEQ,ISET                       SIGD
      IF(IREG3 .GT. IREG3P)GO TO 70                                     SIGD
      IF(IZONE3 .GT. IZON3P)GO TO 70                                    SIGD
      GO TO 450                                                         SIGD
   70 IFILT2 = 0                                                        SIGD
      IF(IZONE3 .EQ. 0)IFILT2=3                                         MOD2
      IF(IREG3 .EQ. 0)IFILT2 = 4                                        SIGD
      IREG3P = IREG3                                                    SIGD
      IZON3P = IZONE3                                                   SIGD
      IF(ISET .NE. 7 .OR. ISEQ .NE. 1)GO TO 450                         SIGD
      READ(ISIGEX,901)ICTRYT,IREGT,IZONET,B1M,B2M,(SIG1M(I),I=1,6),     SIGD
     1(SIG2M(I),I=1,6),ISEQ,ISET                                        SIGD
      IF(ICTRYT .EQ. IZ)GO TO 450                                       SIGD
      IF(ILIST .NE. 0)WRITE(IOUT,903)B1M,B2M,(SIG1M(I),I=1,6),          SIGD
     1(SIG2M(I),I=1,6),ISEQ,ISET                                        SIGD
      IF(IREG3 .NE. IREGT)GO TO 450                                     SIGD
      IF(IZONE3 .NE. IZONET)GO TO 450                                   SIGD
      IF(ISEQ .NE. 2 .OR. ISET .NE. 7)GO TO 450                         SIGD
      READ(ISIGEX,901)ICTRYT,IREGT,IZONET,B10,B20,(SIG10(I),I=1,6),     SIGD
     1(SIG20(I),I=1,6),ISEQ,ISET                                        SIGD
      IF(ICTRYT .EQ. IZ)GO TO 450                                       SIGD
```

```
      IF(ILIST .NE. 0)WRITE(IOUT,903)B10,B20,(SIG10(I),I=1,6),          SIGD
     1(SIG20(I),I=1,6),ISEQ,ISET                                        SIGD
      IF(IREG3 .NE. IREGI)GO TO 450                                     SIGD
      IF(IZONE3 .NE. IZONEI)GO TO 450                                   SIGD
      IF(ISEQ .NE. 3 .OR. ISET .NE. 7)GO TO 450                         SIGD
      READ(ISIGEX,901)ICTRYT,IREGT,IZONET,B1,B2,(SIG1(I),I=1,6),        SIGD
     1(SIG2(I),I=1,6),ISEQ,ISET                                         SIGD
      IF(ICTRYT .EQ. IZ)GO TO 450                                       SIGD
      IF(ILIST .NE. 0)WRITE(IOUT,903) B1,B2,(SIG1(I),I=1,6),(SIG2(I),I= SIGD
     21,6),ISEQ,ISET                                                    SIGD
      IF(IREGT .NE. IREG3)GO TO 450                                     SIGD
      IF(IZONET .NE. IZONE3)GO TO 450                                   SIGD
      IF(ISEQ .NE. 4 .OR. ISET .NE. 7)GO TO 450 .                       SIGD
      ISGCT = ISGCT + 1                                                 SIGD
      IF(B1W .LE. -10. .OR. B1W .GE. 10.)GO TO 200                      SIGD
      IF(B2W .LE. -10. .OR. B2W .GE. 10.)GO TO 200                      SIGD
      IF(B1M .LE. -10. .OR. B1M .GE. 10.)GO TO 200                      SIGD
      IF(B2M .LE. -10. .OR. B2M .GE. 10.)GO TO 200                      SIGD
      IF(B10 .LE. -10. .OR. B10 .GE. 10.)GO TO 200                      SIGD
      IF(B20 .LE. -10. .OR. B20 .GE. 10.)GO TO 200                      SIGD
      IF(B1 .LE. -10. .OR. B1 .GE. 10.)GO TO 200                        SIGD
      IF(B2 .LE. -10. .OR. B2 .GE. 10.)GO TO 200                        SIGD
      GO TO 210                                                         SIGD
  200 IERCDE =-27                                                       SIGD
      CALL ERRPRO                                                       SIGD
  210 DO 240 I=1,6                                                      SIGD
      IF(SIG1W(I) .LT. 0 .OR. SIG1W(I) .GE. 10.)GO TO 250               SIGD
      IF(SIG2W(I) .LT. 0 .OR. SIG2W(I) .GE. 10.)GO TO 250               SIGD
      IF(SIG1M(I) .LT. 0 .OR. SIG1M(I) .GE. 10.)GO TO 250               SIGD
      IF(SIG2M(I) .LT. 0 .OR. SIG2M(I) .GE. 10.)GO TO 250               SIGD
      IF(SIG10(I) .LT. 0 .OR. SIG10(I) .GE. 10.)GO TO 250               SIGD
      IF(SIG20(I) .LT. 0 .OR. SIG20(I) .GE. 10.)GO TO 250               SIGD
      IF(SIG1(I) .LT. 0 .OR. SIG1(I) .GE. 10.)GO TO 250                 SIGD
      IF(SIG2(I) .LT. 0 .OR. SIG2(I) .GE. 10.)GO TO 250                 SIGD
  240 CONTINUE                                                          SIGD
      GO TO 400                                                         SIGD
  250 IERCDE = 28                                                       SIGD
      CALL ERRPRO                                                       SIGD
  900 FORMAT(1H0,49X,11HSIGEXT FILE/3X, 8HICTRY = ,A4, 8H IREG = ,I3,   SIGD
     1 9H IZONE = ,I3/3X,6HB1W = ,F6.3,7H B2W = ,F6.3,9H SIG1W = ,6(F4.2SIGD
     2,1X),9H SIG2W = ,6(F4.2,1X)/3X,6HB1M = ,F6.3,7H B2M = ,F6.3,      SIGD
     39H SIG1M = ,6(F4.2,1X),9H SIG2M = ,6(F4.2,1X)/3X,6HB10 = ,F6.3,   SIGD
     47H B20 = ,F6.3,9H SIG10 = ,6(F4.2,1X),9H SIG20 = ,6(F4.2,1X)/3X,  SIGD
     55HB1 = ,F6.3,6H B2 = ,F6.3,8H SIG1 = ,6(F4.2,1X),8H SIG2 = ,      SIGD
     66(F4.2,1X))                                                       SIGD
  901 FORMAT(4X,A4,I3,I3,2F6.3,12F4.2,4X,2I1)                           SIGD
  902 FORMAT(1H0,45X19HSIG EXT DATA GROUP/15X,4X,A4,I3,I3,2F6.3,12F4.2, SIGD
     14X,2I1)                                                           SIGD
  903 FORMAT(29X,2F6.3,12F4.2,4X,2I1)                                   SIGD
  400 IF(IREG3 .EQ. 0)GO TO 425                                         SIGD
      IF(IREGSV - IREG3)435,405,440                                     SIGD
  405 IF(IZONE3 .EQ. 0)GO TO 425                                        SIGD
      IF(IZONSV - IZONE3)435,425,440                                    SIGD
  425 GO TO 10                                                          SIGD
  435 ISIGE = 1                                                         SIGD
      IF(IZSGE .EQ. 0)GO TO 500                                         SIGD
      IERCDE = 10                                                       SIGD
      CALL ERRPRO                                                       SIGD
      GO TO 500                                                         SIGD
  440 IERCDE = 15                                                       SIGD
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CALL ERRPRO                                                       SIGD
      GO TO 50                                                          SIGD
  450 IERCDE = 2                                                        SIGD
      CALL ERRPRO                                                       SIGD
      GO TO 500                                                         SIGD
  475 ISIGEX = -1                                                       SIGD
      WRITE(ISIGE)IZ,(IZER,I=1,58)                                      SIGD
      REWIND ISIGE                                                      SIGD
  500 RETURN                                                            SIGD
      END
```

@ FLT YESD,1,760407, 58765 , 1

```
      SUBROUTINE YESD                                                  YESD
C     READS A YFS DAT CARD GROUP,CHECKS THE DATA FOR ERRORS AND WRITES YESD
C     A RECORD ON THE YESERR FILE                                      YESD
      COMMON /FLAG/                                                    FLAG
     1ICTRSV,IREGSV,IZONSV,INP,IOUT,ICASE,ILIST,IDERS(2),IOTUS(2),     FLAG
     2IYESU,ISIGEX,ICAMER,ICSFSH,ICTRY,IREG,IZONE,ISTRAT,ISUB,IREGP,   FLAG
     3IZONEP,ISTRAP,IFATER,IZSGC,ITYPE,IFATEY,IFATEC,IFATES,IRANGE,    FLAG
     4ICAMR,IYESR,ISICR,IFOR,IPRINT,ICFSB,ISBHS,IFRCOF,ISUBCI,         FLAG
     5IYSCI,ICMCI,ISGCI,IYSERC,ICMERC,ISIGEC,IPSEC,IYESER,ICAME,       FLAG
     6ISIGL,IREG2P,IZON2P,ISTR2P,ISUB2P,ISEG2P,IREG1P,IZON1P,ISTR1P,   FLAG
     7IREG3P,IZON3P,ILEVEL,IFILT1,IFILT2                               FLAG
     8,ILEVES                                                          MOD3
      COMMON /YESIN/                                                   YESIN
     1ICTRY1,IREG1,IZONE1,ISTRA1,YIELDI,IERDTE(3,6),BIASYI(6),SIGYII(6),YESIN
     2YIELD,IRDTE(6),BIASY(6),SIGYIE(6)                                YESIN
      DIMENSION IDIEM(6)                                               YESD
      DATA IZ/4HZZZZ/,IZER/0/,ICON/4HUSA /,ICON1/4HCAND/               YESD
      ITYPE = 1                                                        YESD
      IF(ICTRY .EQ. ICON .OR. ICTRY .EQ. ICON1)GO TO 2                 YESD
      XCON = 100.                                                      YESD
      XCOM = -XCON                                                     YESD
      XCONV = 1.0                                                      MOD7
      GO TO 5                                                          YESD
    2 XCON = 100.*1.4869664                                            MOD7
      XCOM = -XCON                                                     YESD
      XCONV = 1.4869664                                                MOD7
    5 IF(IYESR .EQ. 0)GO TO 10                                         YESD
      IYESR = 0                                                        YESD
      GO TO 200                                                        YESD
   10 READ(IYESU,900)ICTRY1,IREG1,IZONE1,ISTRA1,YIELDI,((IERDTE(I,J),I=1YESD
     1,3),J=1,6),ISEQ,ISET                                             YESD
      IF(ICTRY1 .EQ. IZ)GO TO 475                                      YESD
      IF(ILIST .NE. 0)WRITE(IOUT,901)ICTRY1,IREG1,IZONE1,ISTRA1,YIELDI, YESD
     1((IERDTE(I,J),I=1,3),J=1,6),ISEQ,ISET                            YESD
      IF(IREG1 .GT. IREG1P)GO TO 20                                    YESD
      IF(IZONE1 .GT. IZON1P)GO TO 20                                   YESD
      IF(ISTRA1 .GT. ISTR1P)GO TO 20                                   YESD
      IF(IZONE1 .GT. IZON1P)GO TO 20                                   YESD
      IF(ISTRA1 .GT. ISTR1P)GO TO 20                                   YESD
      GO TO 450                                                        YESD
   20 IF(ISEQ .NE. 1 .OR. ISET .NE.5)GO TO 450                         YESD
      IREG1P = IREG1                                                   YESD
      IZON1P = IZONE1                                                  YESD
      ISTR1P = ISTRA1                                                  YESD
      READ(IYESU,902)ICTRY1,IREG1,IZONE1,ISTRA1,(BIASYI(I),I=1,6),     YESD
     1(SIGYII(I),I=1,6),ISEQ,ISET                                      YESD
      IF(ICTRY1 .EQ. IZ)GO TO 450                                      YESD
      IF(ILIST .NE. 0)WRITE(IOUT,903)(BIASYI(I),I=1,6),(SIGYII(I),I=1,6)YESD
     1,ISEQ,ISET                                                       YESD
      IF(ISEQ .NE. 2 .OR. ISET .NE. 5)GO TO 450                        YESD
      IF(IREG1 .NE. IREG1P)GO TO 450                                   YESD
      IF(IZONE1 .NE. IZON1P)GO TO 450                                  YESD
      IF(ISTRA1 .NE. ISTR1P)GO TO 450                                  YESD
      IYSCI = IYSCI + 1                                                YESD
      IPRINT = 0                                                       YESD
      IF(IERDTE(1,1) .EQ. 0)GO TO 190                                  YESD
      DO 30 I=1,6                                                      YESD
      IF(IERDTE(1,I) .EQ. 0)GO TO 26                                   MOD3
```

```
      IF(IERDTE(1,I) .GT. 60)GO TO 22                                  MOD3
      GO TO 32                                                         MOD3
   22 IF(IERDTE(2,I) .GT. 0 .AND. IERDTE(2,I) .LE. 12)GO TO 24         MOD3
      GO TO 32                                                         MOD3
   24 IF(IERDTE(3,I) .GT. 0 .AND. IERDTE(3,I) .LE. 31)GO TO 26         MOD3
      GO TO 32                                                         MOD3
   26 FLDA = IERDTE(3,I)                                               MOD3
      IDTEM(I) = 0                                                     MOD1
      IF(FLDA .EQ. 0.0)GO TO 30                                        MOD1
      CALL IFPA(FLDA ,IERDTE(2,I),IERDTE(1,I),DAYS)                    YESD
      IDTEM(I)  = DAYS                                                 YESD
   30 CONTINUE                                                         YESD
      GO TO 34                                                         MOD3
   32 IERCDE = 25                                                      MOD3
      CALL ERRPRO                                                      MOD3
      GO TO 55                                                         MOD3
   34 DO 35 I=1,5                                                      MOD3
      IF(IDTEM(I+1) .EQ. 0)GO TO 55                                    MOD1
      IF(IDTEM(I) .LT. IDTEM(I+1))GO TO 35                             YESD
      GO TO 40                                                         YESD
   35 CONTINUE                                                         YESD
      GO TO 55                                                         YESD
   40 IERCDE = 20                                                      YESD
      CALL ERRPRO                                                      YESD
   55 DO 60 I=1,6                                                      YESD
      IF(SIGYII(I) .LT. 0 .OR. SIGYII(I) .GE. XCON)GO TO 65            YESD
   60 CONTINUE                                                         YESD
      GO TO 70                                                         YESD
   65 IERCDE = 23                                                      YESD
      CALL ERRPRO                                                      YESD
   70 DO 75 I=1,6                                                      YESD
      IF(BIASYI(I) .LE. XCOM .OR. BIASYI(I) .GE. XCON)GO TO 80         YESD
   75 CONTINUE                                                         YESD
      GO TO 190                                                        YESD
   80 IERCDE = 22                                                      YESD
      CALL ERRPRO                                                      YESD
  190 IF(YIELDI .LT. 0 .OR. YIELDI .GE. XCON)GO TO 195                 YESD
      GO TO 200                                                        YESD
  195 IERCDE = 21                                                      YESD
      CALL ERRPRO                                                      YESD
  200 IF(IREG - IREG1)240,205,250                                      YESD
  205 IF(IZONE - IZONE1)240,210,250                                    YESD
  210 IF(ISTRAT - ISTRAT)240,215,250                                   YESD
  215 IF(IFATFY .NE. 0)GO TO 500                                       YESD
      YIELD = YIELDI/XCONV                                             MOD7
      IF(IERDTE(1,1) .EQ. 0)GO TO 230                                  YESD
      DO 220 I=1,6                                                     YESD
      ITRDTE(I) = IDTEM(I)                                             YESD
      BIASY(I) = BIASYI(I)/XCONV                                       MOD7
      SIGYIE(I) = SIGYII(I)/XCONV                                      MOD7
  220 CONTINUE                                                         YESD
  230 IF(ILIST .EQ. 2)WRITE(IOUT,904)ICTRY,IREG,IZONE,ISTRAT,YIELD,    YESD
     1(ITRDTE(I),BIASY(I),SIGYIE(I),I=1,6)                             YESD
      WRITE(IYESER)ICTRY,IREG,IZONE,ISTRAT,YIELD,(ITRDTE(I),BIASY(I),  YESD
     1SIGYIE(I),I=1,6)                                                 YESD
      IYSERC = IYSERC + 1                                              YESD
      GO TO 600                                                        YESD
  240 IYESR = 1                                                        YESD
      IERCDE = 10                                                      YESD
      CALL ERRPRO                                                      YESD
```

```
      GO TO 500                                                         YESD
  250 IERCDE = 15                                                       YESD
      CALL ERRPRO                                                       YESD
      GO TO 5                                                           YESD
  450 IERCDE = 2                                                        YESD
      CALL ERRPRO                                                       YESD
      GO TO 500                                                         YESD
  475 IYESD = -1                                                        YESD
      WRITE(IYESER)IZ,(IZER,I=1,22)                                     YESD
      REWIND IYESER                                                     YESD
  500 RETURN                                                            YESD
  900 FORMAT(4X,A4,2I3,I4,F5.2,18I2,19X,2I1)                            YESD
  901 FORMAT(1H0,4/X,14HYES DATA GROUP/15X,4X,A4,2I3,I4,F5.2,18I2,19X,  YESD
     12I1)                                                              YESD
  902 FORMAT(4X,A4,2I3,I4,6F5.1,6F5.2,2I1)                              YESD
  903 FORMAT(33X,6F5.1,6F5.2,2I1)                                       YESD
  904 FORMAT(50X,11HYESERR FILE/3X,8HICTRY = ,A4, 8H IREG =  ,I3, 9H IZOYESD
     1NE = ,I3, 9H ISTRAT =,I4/3X,4HY = ,F5.2,40H (ITRDTE(I),9IASY(I),SIYESD
     2GYIF(I),I=1,6) = /3X,6(I5,1X,F5.1,1X,F5.2,1X))                    YESD
       END                                                              YESD
```

15. TOC